Suivi tridimensionnel de la main et reconnaissance de gestes pour les Interfaces Homme Machine

Simon Conseil

UNIVERSITÉ PAUL CÉZANNE AIX-MARSEILLE III

N° attribué par la bibliothèque : 2008AIX30006

SUIVI TRIDIMENSIONNEL DE LA MAIN ET RECONNAISSANCE DE GESTES POUR LES INTERFACES HOMME-MACHINE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ PAUL CÉZANNE
FACULTÉ DES SCIENCES ET TECHNIQUES

Discipline : Optique, électromagnétique et image

présentée et soutenue publiquement par :

Simon CONSEIL

Directeur de Thèse : Pr. Salah BOURENNANE

École doctorale : Physique et Sciences de la Matière

Soutenue publiquement le 13 mars 2008
devant le jury composé de :

RAPPORTEURS :	Alice CAPLIER	MCF HDR, GIPSA-lab, INP Grenoble
	Liming CHEN	Pr., LIRIS, École Centrale Lyon
EXAMINATEURS :	Monique THONNAT	DR, INRIA Sophia Antipolis
	Salah BOURENNANE	Pr., Institut Fresnel, École Centrale Marseille
	Stéphane DERRODE	MCF, Institut Fresnel, École Centrale Marseille
	Lionel MARTIN	Ingénieur de recherche, STMicroelectronics

ANNÉE : 2008

Simon Conseil : *Suivi tridimensionnel de la main et reconnaissance de gestes pour les Interfaces Homme-Machine,* Thèse de doctorat, Université Paul Cézanne Aix-Marseille III, © 2008

La science est ce que nous comprenons suffisamment bien pour l'expliquer à un ordinateur. L'art, c'est tout ce que nous faisons d'autre[1].

— Donald Knuth

REMERCIEMENTS

Tout d'abord, je remercie Salah Bourennane pour m'avoir permis de faire cette thèse et pour la confiance qu'il m'a accordé. Je remercie également Lionel Martin, pour sa disponibilité de tous les instants et pour sa bonne humeur. Merci à Frédéric Guérault et à Christophe Chesnaud, pour m'avoir ouvert les portes de 3DFeel. Enfin, je tiens à remercier Stéphane Derrode pour l'aide et les conseils qu'il m'a apporté lors de la rédaction de cette thèse.

Je voudrais également adresser mes plus vifs remerciements à chacun des membres du jury : Monique Thonnat, pour m'avoir fait l'honneur de présider mon jury de thèse, Alice Caplier et Liming Chen, pour avoir accepté la charge de rapporter cette thèse. Merci pour l'intérêt qu'ils y ont porté et pour leurs remarques.

Merci aux collègues de l'Institut Fresnel et aux membres de l'équipe GSM, permanents, doctorants et stagiaires avec qui j'ai partagé de bon moments pendant ces quatre années. Merci tout particulièrement à Nadine, Cyril, Damien et William pour leur bonne humeur, leur support et leurs encouragements.

Merci aux collègues de STMicroelectronics pour leur accueil chaleureux au sein de l'équipe AST et pour la bonne ambiance de travail. Merci tout particulièrement à Sophie et Réouven.

Je remercie bien sûr mes parents et toute ma famille, qui m'ont toujours soutenu dans mes choix et m'ont permis d'en arriver là.

Merci enfin à Stéphanie, pour m'avoir supporté et soutenu durant la rédaction de cette thèse.

1. *Science is what we understand well enough to explain to a computer. Art is everything else we do.*

SOMMAIRE

ABRÉVIATIONS

Pour des raisons de lisibilité, la signification d'une abréviation ou d'un acronyme n'est généralement rappelée qu'à sa première apparition, en note de bas de page. Par ailleurs, puisque nous utilisons toujours l'abréviation la plus usuelle, il est fréquent que ce soit le terme anglais qui soit employé. Dans ce cas, nous présentons une traduction.

GUI	*Graphical User Interface* (interface graphique)
PUI	*Perceptual User Interface* (interface perceptuelle)
WIMP	*Window, Icon, Menu, Pointing device* (fenêtre, icône, menu, dispositif de pointage)
ACP	Analyse en Composantes Principales
ASL	*American Sign Language* (langue des signes américaine)
DTW	*Dynamic Time Warping* (recalage dynamique)
EM	*Expectation-Maximisation* (espérance-maximisation)
FFT	*Fast Fourier Transform* (Transformée de Fourier Rapide)
IHM	Interface Homme Machine
LPC	Langage Parlé Complété
LSF	Langue des Signes Française
HMM	*Hidden Markov Models* (Modèles de Markov Cachés)
k-NN	*k-Nearest Neighbors* (*k*-plus proches voisins)
RBF	*Radial-Basis Function* (fonctions à base radiale)
SVM	*Support Vector Machine* (machine à support vectoriel)
CMC	Articulation carpo-métacarpienne
IP	Articulation inter-phalangienne
IPD	Articulation inter-phalangienne distale
IPP	Articulation inter-phalangienne proximale
MCP	Articulation métacarpo-phalangienne

1 INTRODUCTION

Le sujet de nos travaux de recherche concerne la conception et le développement de méthodes de vision par ordinateur pour la reconnaissance de gestes de la main. Nous cherchons à répondre aux besoins de conception d'une Interface Homme-Machine dont l'objectif est de transformer un écran classique en surface tactile et de permettre à l'utilisateur de se servir de son doigt comme dispositif de pointage.

Les gestes de la main sont un canal de communication naturel et intuitif chez l'homme pour interagir avec son environnement. Ils servent à désigner ou à manipuler des objets, à renforcer la parole, ou à communiquer basiquement dans un environnement bruité. Ils peuvent aussi représenter un langage à part entière avec la langue des signes [102]. Les gestes peuvent avoir une signification différente suivant la langue ou la culture : les langues des signes en particulier sont spécifiques à chaque langue.

Pour Cadoz [17], le geste est un des canaux de communications les plus riches. Ainsi, dans le domaine des *Interfaces Homme-Machine* (IHM), la main peut servir à pointer (pour remplacer la souris), à manipuler des objets (pour la réalité augmentée ou virtuelle), ou à communiquer par gestes avec un ordinateur. Par rapport à la richesse de l'information véhiculée par les gestes de la main, les possibilités de communication avec les ordinateurs sont aujourd'hui réduites avec la souris et le clavier. L'interaction homme-machine est basée actuellement sur le paradigme WIMP [1] qui présente les bases fonctionnelles d'une interface *graphique* informatique (GUI [2]). La majorité des systèmes d'exploitation repose sur ce concept, avec un dispositif de pointage, généralement la souris, qui permet d'interagir avec des éléments graphiques tels que des fenêtres, des icônes et des menus, de façon plus intuitive qu'avec une interface *textuelle* (en ligne de commande). En utilisant les gestes de la main, l'interface devient *perceptuelle* (PUI [3]).

Les systèmes de reconnaissance de gestes ont d'abord utilisé des gants électroniques munis de capteurs fournissant la position de la main et les angles des articulations des doigts [12]. Mais ces gants sont onéreux et encombrants, d'où l'intérêt croissant pour les méthodes de vision par ordinateur. En effet, avec les progrès techniques et l'apparition de caméras bon marché, il est désormais possible de développer des systèmes de reconnaissance de gestes basés sur la vision par ordinateur, fonctionnant en temps réel [105].

Toutefois, la main étant un organe complexe, déformable, comportant de nombreux degrés de liberté au niveau des articulations, il est difficile de reconnaître sa forme à partir d'images sans un certain nombre de contraintes et d'a priori. En effet, l'homme peut effectuer naturellement un très grand nombre de gestes différents.

Avec l'évolution des technologies d'acquisition et des techniques de reconnaissance de gestes, de nombreux domaines d'application ont émergé :

– la *reconnaissance de la langue des signes* [12, 32, 121] ;

1. *Window, Icon, Menu, Pointing device*
2. *Graphical User Interface*
3. *Perceptual User Interface*

- la *réalité virtuelle*, où la main sert à manipuler des objets virtuels et déclencher des actions, ou à naviguer dans un environnement virtuel [63, 114] ;
- la *réalité augmentée*, où le monde physique est augmenté avec des informations virtuelles, par exemple par une rétro-projection [84] ;
- les *applications multimodales*, associant le geste à d'autres moyens de communication, tels que la parole ou les expressions du visage [131] ;
- le codage et la transmission de gestes à bas débit pour la télé-conférence [2, 33] ;
- la *biométrie*, pour la reconnaissance de personnes avec la forme de leur main [82, 141].

1.1 LES GESTES DE LA MAIN

Une réflexion sur les gestes à utiliser est nécessaire, afin que les utilisateurs puissent les réaliser intuitivement, ou avec une période d'apprentissage limitée. Quels gestes faut-il utiliser ? Sont-ils faciles à reproduire ? À quelles actions sont-ils intuitivement associés ?

D'une manière générale, le geste est assimilé à tous les mouvements d'une partie du corps.Le geste de la main est à la fois un moyen d'action, de perception et de communication [17]. Les différentes fonctions du canal gestuel sont ici présentées succinctement. Pour plus de détails sur le sujet, nous renvoyons le lecteur aux thèses de BRAFFORT [12] et de MARTIN [92], qui proposent des études détaillées du canal gestuel.

Les trois fonctions du geste humain

Les gestes sont un des moyens de communication les plus riches que l'être humain possède. Ils permettent d'agir sur le monde physique, et servent aussi à communiquer. De plus, le geste permet à la fois d'émettre des informations, mais aussi d'en recevoir.

CADOZ [17] définit trois fonctions principales de la main :

LA FONCTION ERGOTIQUE : la main joue le rôle d'organe moteur et agit sur le monde physique pour le transformer. Elle applique aux objets des forces, pour les déplacer ou les déformer.

LA FONCTION ÉPISTÉMIQUE : la main joue le rôle d'organe de perception. Le sens du *toucher* (sens *tactilo-proprio-kinesthésique*) donne des informations sur la forme, l'orientation, la distance, la grandeur, le poids, la température, les mouvements des objets, etc.

LA FONCTION SÉMIOTIQUE : la main joue le rôle d'organe d'expression pour l'émission d'informations visuelles. Cela comprend la langue des signes, le geste co-verbal, qui accompagne la parole, ou les gestes permettant une communication basique lorsqu'on ne peut pas utiliser la parole, comme dans un environnement bruité ou en plongée sous-marine.

Dans le cadre de la reconnaissance de gestes pour les IHM, nous nous intéressons plus particulièrement à la fonction sémiotique. Cette fonction est la plus riche et la plus complexe. Elle peut être décomposée en plusieurs catégories, et différentes classifications ont été proposées.

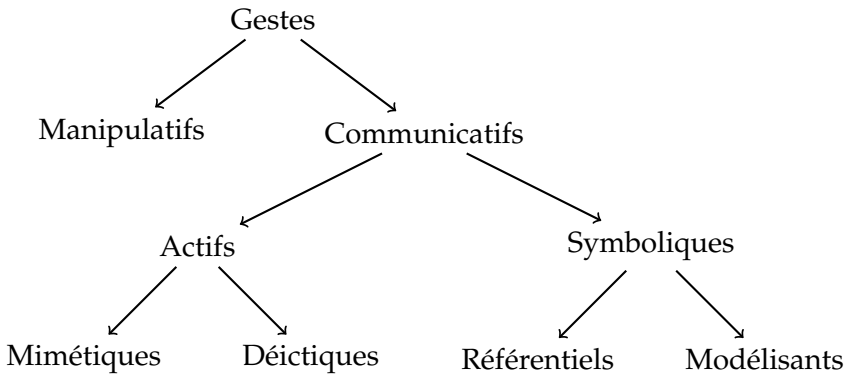


FIGURE 1.1 – *Taxonomie des gestes de* QUEK [110].

Taxonomie des gestes de QUEK

Une classification bien adaptée au domaine de l'interaction homme-machine est la taxonomie de QUEK [110], qui décompose les gestes en gestes *manipulatifs*, correspondant aux fonctions *ergotique* et *épistémique* ; et en gestes *communicatifs*, correspondant à la fonction *sémiotique*. Les gestes *communicatifs* sont décomposés en gestes *actifs* et en gestes *symboliques* (figure 1.1).

Les gestes *symboliques* ne sont pas compréhensibles directement, il faut être initié pour comprendre leur signification. Il s'agit par exemple des gestes des langues des signes. Ils sont décomposés en gestes *référentiels*, faisant directement référence à un objet ou à un concept (p. ex. le frottement du pouce et de l'index pour évoquer l'argent), et en gestes *modélisants*, qui modélisent un état ou une opinion et s'emploient souvent en addition à d'autres moyen de communication (p. ex. pour donner une idée de la taille d'un objet). Ces gestes peuvent avoir un sens différent selon la culture.

Les gestes *actifs* sont directement liés à leur interprétation et sont utilisés en complément de la parole. Ils sont décomposés en gestes *mimétiques*, consistant à mimer une action, et en gestes *déictiques*, ou gestes de pointage. Les gestes déictiques sont très utilisés pour l'interaction homme-machine, car le doigt représente un dispositif de pointage naturel et très pratique.

Gestes statiques et dynamiques

Il existe deux autres catégories de gestes : les gestes *statiques*, ou postures, et les gestes *dynamiques* (figure 1.2). Par ailleurs, il faut distinguer la position et la configuration de la main. En combinant ces deux aspects, on obtient la classification proposée par HARLING ET EDWARDS [54] :

- position statique, configuration statique (les postures) ;
- position statique, configuration dynamique ;
- position dynamique, configuration statique (p. ex. les gestes de pointage) ;
- position dynamique, configuration dynamique (p. ex. la langue des signes).

Mais il est aussi possible de regrouper la position et la configuration de la main dans un vecteur de mesures à n dimensions. Ainsi, un geste est représenté par une trajectoire dans l'espace des mesures (MARTIN [92]). PAVLOVIC *et al.* [105] propose la définition suivante du geste : « Un geste de la main est un processus

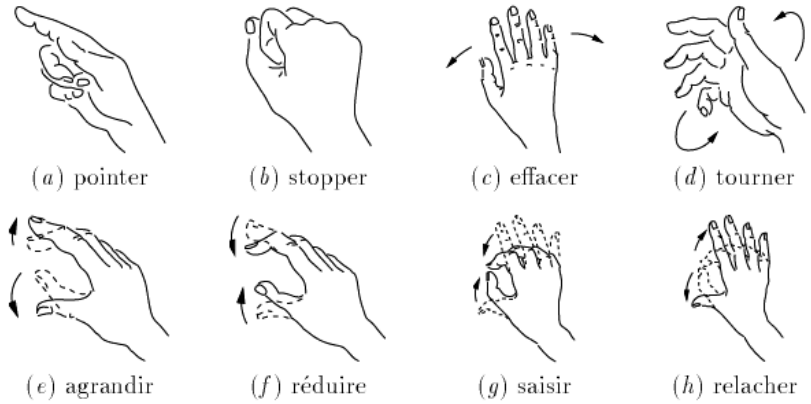


FIGURE 1.2 – *Exemples de gestes statiques et dynamiques, utilisés par* QUEK [110] *(les pointillés représentent la configuration initiale), extrait de* MARTIN [92].

stochastique dans l'espace paramétrique des gestes sur un interval de temps déterminé. Deux réalisations d'un même geste ne donnent pas exactement le même vecteur de paramètres, mais les valeurs sont suffisamment proches pour que le geste soit identifiable.

1.2 SUJET DE RECHERCHE

Nous cherchons à concevoir des méthodes de vision par ordinateur répondant à des critères précis, dans un contexte appliqué. Notre approche est dirigée par une conception de type « *top-down* », consistant d'abord à identifier les besoins liés à notre application, et ensuite à concevoir des techniques adaptées, répondant à ces besoins. En effet, la vision par ordinateur offre de nombreuses possibilités, mais certaines solutions ne sont pas adaptées à notre application, notamment du fait d'un manque de robustesse aux conditions réelles ou d'une complexité trop importante pour être mise en oeuvre en temps réel.

L'application directe de nos travaux et le contexte industriel (détaillé dans le chapitre 2) nous imposent de considérer les contraintes suivantes :

- les caméras sont de types « *webcams* » à faible coût,
- les traitements doivent être temps réel,
- les méthodes doivent être robustes aux conditions d'acquisition,
- les contraintes imposées aux utilisateurs doivent être minimales.

Une étude de la littérature sur le domaine nous permet d'analyser les différentes approches et de choisir une approche par apparence, plus adaptée à notre application qu'une approche par modèle 3D. Nous proposons alors des techniques pour mettre en oeuvre les différentes étapes d'un système de reconnaissance de gestes, décomposées selon le schéma suivant :

- segmentation de la main dans le flux vidéo ;
- extraction de caractéristiques représentant la configuration de la main ;
- reconnaissance de gestes parmi un ensemble prédéfini (vocabulaire) ;
- suivi de la main en 2D avec une caméra, et en 3D avec deux caméras, en vision stéréoscopique passive.

Ces étapes correspondent aux chapitres de ce manuscrit et sont détaillées dans la section suivante.

1.3 ORGANISATION DU MANUSCRIT

LE CHAPITRE 2 présente le contexte industriel de cette thèse, ainsi que la configuration expérimentale que nous avons choisi (type et position des caméras, gestes utilisés). Les différentes données, utilisées dans la suite pour évaluer nos algorithmes, sont ensuite présentées.

LE CHAPITRE 3 présente le domaine de l'interaction homme-machine gestuelle. Les dispositifs classiques d'interaction sont étudiés, ainsi que les nouvelles possibilités d'interaction. Les nombreuses applications sont détaillées, de la réalité augmentée ou virtuelle à la reconnaissance de la langue des signes. Nous présentons ensuite un état de l'art des approches basées sur la vision par ordinateur, dans le contexte des IHM. L'interprétation visuelle des gestes offre l'interaction la plus naturelle et intuitive, mais elle est aussi la plus difficile à mettre en oeuvre. De nombreuses approches ont été proposées pour reconnaître les gestes avec des caméras. Des compléments de bibliographie sont apportés dans les chapitres suivants.

LE CHAPITRE 4 présente la segmentation de la main dans un flux vidéo, qui est une étape déterminante pour la suite des traitements. Nous présentons une approche basée sur une modélisation de la couleur de peau par un histogramme adaptatif dans l'espace C_bC_r. Afin d'éviter un apprentissage hors-ligne, l'histogramme est initialisé automatiquement en utilisant une détection par des seuils dans l'espace C_bC_r. Nous présentons ensuite l'extraction de caractéristiques morphologiques, à partir de l'image binaire et du contour de la main. Nous proposons des méthodes pour extraire des caractéristiques de position (centre de la main, poignet, et bouts des doigts), qui sont utilisées pour le suivi des doigts et de la main (chapitre 6).

LE CHAPITRE 5 présente une comparaison de descripteurs de formes pour la reconnaissance de gestes. Pour évaluer les descripteurs (invariants de HU, moments de ZERNIKE, et descripteurs de FOURIER) et leurs propriétés d'invariance aux transformations euclidiennes, nous avons défini un vocabulaire de gestes et construit une base de données d'images. Les gestes à reconnaître sont classifiés par rapport aux modèles calculés lors d'une étape d'apprentissage. Dans un deuxième temps, nous nous intéressons à la reconnaissance dans un flux vidéo.

LE CHAPITRE 6 présente une méthode de suivi des doigts et de la main en vision stéréoscopique. Le filtre de KALMAN permet d'estimer la position 3D de chaque doigt avec des observations bruitées. Cette méthode est d'abord appliquée au geste de pointage, avant d'être étendue au suivi multi-doigts. Nous présentons ensuite une méthode de suivi 2D de la main, basée sur un modèle squelettique permettant d'améliorer la robustesse du suivi et de connaître la position de chaque doigt au fil du temps. Le modèle est recalé dans les images avec des points caractéristiques tels que le centre de la main et les bouts des doigts. Cette approche est étendue en 3D, en combinant le modèle estimé dans les deux vues et en visualisant le résultat avec un modèle 3D. Ce dernier permet également de prendre en compte des contraintes supplémentaires sur la morphologie de la main.

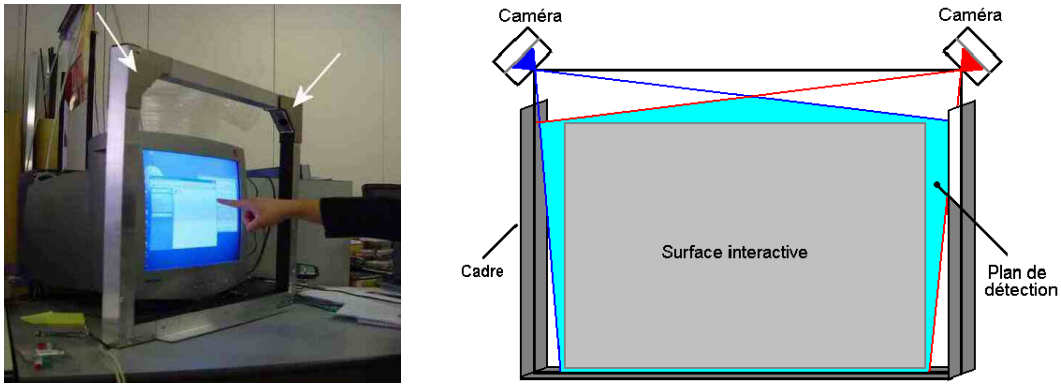
2

CONTEXTE INDUSTRIEL ET CONFIGURATION EXPÉRIMENTALE

Ce chapitre présente tout d'abord la collaboration industrielle à l'origine de cette thèse. Ce contexte a fortement influencé les choix techniques, notamment pour les caméras, et pour les contraintes de temps réel et de robustesse des algorithmes. Les gestes auxquels nous nous intéressons sont choisis en fonction de l'application visée.

Après une présentation du système `3DFeel`, nous présentons notre configuration expérimentale ainsi que les caméras utilisées. Nous décrivons enfin les gestes considérés et les différentes données (séquences vidéos et bases d'images) utilisées dans la suite pour évaluer nos algorithmes.

SOMMAIRE



(a) (b)

FIGURE 2.1 – *Le système* `3DFeel` *(© 3DFEEL).*

2.1 CONTEXTE INDUSTRIEL DE LA THÈSE

Le travail de thèse s'est déroulé dans le cadre d'un programme de recherche financé par la région PACA [1], le conseil général des Bouches-du-Rhône et la communauté du pays d'Aix-en-Provence, visant à développer les collaborations entre les laboratoires et les entreprises de la région. Le projet lié à cette thèse s'intitule « *Interface Homme-Machine* 3D *pour un environnement intelligent* », et implique trois partenaires :

- l'INSTITUT FRESNEL (CNRS UMR 6133),
- STMICROELECTRONICS [2],
- 3DFEEL [3].

Ce projet est basé sur le système développé par l'entreprise 3DFEEL : une interface utilisant la vision par ordinateur, et permettant de transformer un écran classique en écran tactile. Un des objectifs du projet est l'intégration des traitements « bas niveau » au plus proche des capteurs CMOS, technologie développée par STMICROELECTRONICS.

La collaboration industrielle à l'origine de cette thèse a donc guidé les choix sur le matériel utilisé, et sur les contraintes de temps réel et de robustesse des algorithmes.

Le système `3DFeel`

Le système `3DFeel` (figure 2.1a) permet de rendre une surface interactive en utilisant deux caméras CMOS (pointée par les flèches sur la figure). Un cadre est utilisé pour délimiter la zone interactive et détecter la présence d'un doigt. La robustesse du système aux variations de luminosité est améliorée par un système d'éclairage, intégré au cadre, dans le proche infrarouge. De plus, les deux caméras sont équipées d'un jeu de filtres permettant de limiter leur sensibilité au domaine spectral d'émission de l'éclairage infrarouge.

Le cadre joue un rôle essentiel dans la détection. Sa principale caractéristique est qu'il doit rester suffisamment noir sur les images quelque soit la puissance de

1. Provence-Alpes-Côte d'Azur
2. http://www.st.com
3. http://www.3dfeel.com

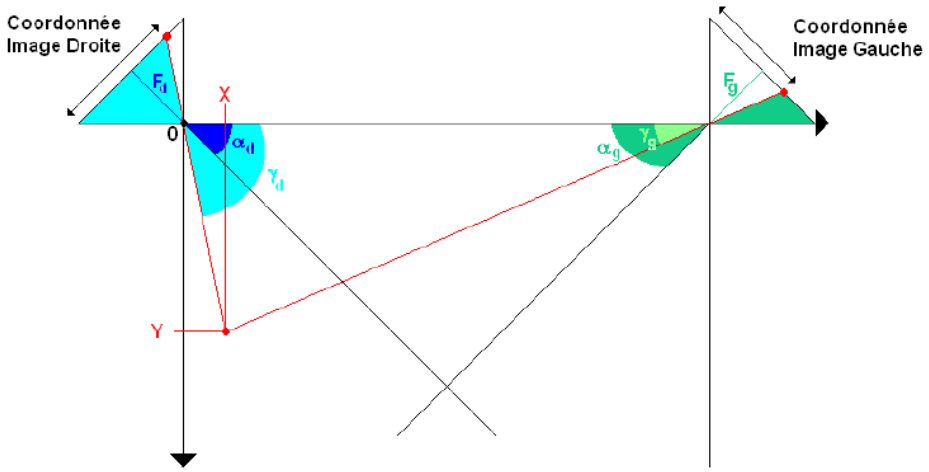


FIGURE 2.2 – *Calcul par triangulation de la position du doigt dans la surface, à partir des détections dans les deux caméras (© 3DFEEL).*

l'éclairage ambiant. Il ne doit ni saturer les capteurs sous l'éclairage infrarouge, ni sous la lumière solaire. Cette caractéristique permet de détecter aisément un doigt présent à l'intérieur du cadre, grâce au contraste élevé entre le doigt clair et le cadre sombre.

Les caméras sont disposées aux deux coins supérieurs du cadre (figure 2.1), de telle sorte que le plan de détection formé par le cadre soit inclus dans le champ de vision de chacune des caméras. Afin d'obtenir cette propriété, les capteurs sont munis d'objectifs grand angle (de 60° à 90°).

Pour chaque caméra, une région d'intérêt (ROI [4]) est définie. L'algorithme de détection du doigt ne s'applique sur l'image qu'à l'intérieur de cette ROI. Cet algorithme utilise les pixels de la ROI pour calculer un signal. Le doigt correspond à un pic dans ce signal. Les coordonnées du doigt détecté dans chaque caméra permettent de calculer la position du bout du doigt par triangulation (figure 2.2). Les positions calculées sont ensuite filtrées afin de lisser la trajectoire du doigt. Les coordonnées obtenues peuvent alors être utilisées pour remplacer la souris d'un ordinateur.

Les principales applications consistent à transformer une surface de projection ou un écran en une surface tactile, permettant à l'utilisateur de se servir de son doigt comme dispositif de pointage. La technologie `3Dfeel` peut transformer un écran standard (en particulier un écran d'ordinateur) en un véritable écran interactif. Les fonctionnalités de cet écran dépassent celles des écrans tactiles standard. Les écrans tactiles souffrent en effet de certaines limitations : ils sont onéreux, limités en surface, sujet à l'usure et sensibles aux rayures (suivant la technologie employée, cf. paragraphe 3.1.1.3).

À l'inverse, la technologie `3Dfeel` est peu coûteuse, grâce à l'utilisation de caméras CMOS à faible coût, et l'utilisateur n'a pas besoin de toucher l'écran, ce qui limite l'usure de celui-ci. De plus, cette technologie peut être adaptée à des surfaces importantes. Ainsi, `Ubiq'window` est un autre système développé par 3DFEEL, utilisant la même technologie et permettant une interaction avec tout type d'élément, quelque soit sa nature ou sa dimension (mur, vitrine, mobilier, linéaire de magasin...).

4. *Region Of Interest*

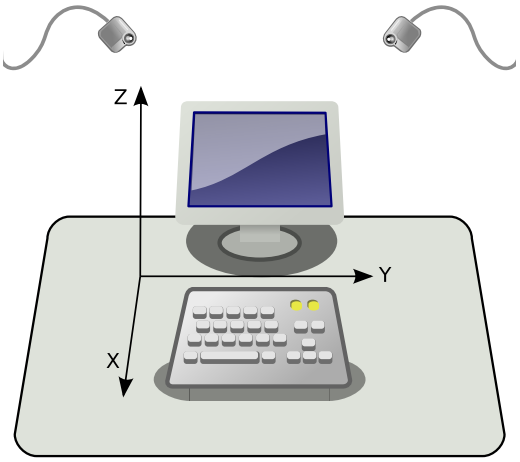


FIGURE 2.3 – *Notre configuration.*

Le démonstrateur présenté sur la figure 2.1a est encombrant, mais des possibilités d'intégration des caméras au plus près d'un écran TFT ont été étudiées par 3DFEEL. Par ailleurs, STMICROELECTRONICS a travaillé sur l'intégration de l'algorithme de détection du doigt sur une carte FPGA [5], avec pour objectif de réaliser les traitements bas-niveaux sur un co-processeur et de n'envoyer à l'ordinateur que la position du doigt.

2.2 CONFIGURATION EXPÉRIMENTALE

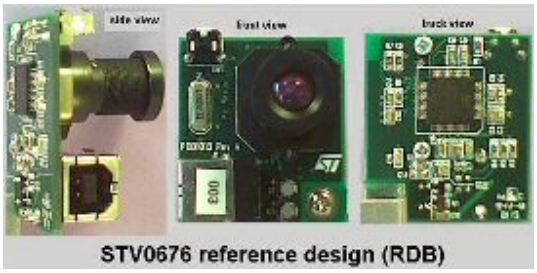
Notre configuration expérimentale est directement inspirée du système `3DFeel`. L'objectif est d'étendre la zone d'interaction en 3D avec la vision stéréoscopique, et d'offrir de nouvelles possibilités d'interactions grâce à la reconnaissance automatique de gestes.

Le système `3DFeel` est prometteur mais sa conception même fait que son champ d'interaction est limité :

- les gestes qui peuvent être reconnus sont limités à des trajectoires d'un ou plusieurs doigts ;
- la surface d'interaction est limitée par le cadre, ce qui restreint la dimension des mouvements que l'utilisateur peut effectuer, et la proximité des caméras pose des problèmes de distorsion sur les bords de cette zone ;
- l'éclairage infrarouge facilite la détection de la main, mais nécessite une source d'éclairage supplémentaire, et il n'est pas possible d'utiliser l'information de couleur ;
- l'angle et la distance entre les caméras sont importants. Il n'est donc pas possible de rectifier les images pour calculer une carte de profondeur (cf. paragraphe B.3.4), par contre cette configuration permet une meilleure précision au niveau de la triangulation 3D.

Dans la continuité du système `3DFeel`, notre idée consiste à disposer les deux caméras au-dessus du clavier d'un ordinateur (figure 2.3), afin de permettre une interaction 3D dans le volume délimité par les caméras, l'écran et le bureau. L'utilisateur peut ainsi interagir avec son ordinateur, et avoir un retour direct d'informations sur l'écran. Grâce à la miniaturisation des caméras, il est

5. *Field-Programmable Gate Array.*



(a) *STV*0676 (b) *VS*6524

FIGURE 2.4 – *Les caméras produites par* STMICROELECTRONICS.

envisageable d'intégrer ce dispositif sur l'écran (cet aspect du projet n'est pas développé ici).

Notre système repose sur l'utilisation de deux caméras (présentées dans la section 2.3), du même type que celles utilisées dans le système `3Dfeel`. Les caméras sont positionnées au-dessus du bureau, de manière à observer le bureau, l'écran, et le volume 3D correspondant. L'axe optique des caméras est pratiquement perpendiculaire au plan du bureau. Les caméras se trouvent donc dans un champ proche de la main. Pour avoir un champ de vision suffisamment important, nous utilisons des optiques avec un grand angle de vue (90°), ce qui provoque des distorsions dans les images. De plus, chaque caméra est connectée sur un port USB. Les caméras ne peuvent donc pas être synchronisées, ce qui provoque des difficultés au niveau de la reconstruction 3D.

Avec deux caméras, la vision stéréoscopique permet de calculer la position 3D à partir d'une paire de points appariés. Une étape de calibration est réalisée pour obtenir une reconstruction 3D euclidienne (cf. section B.2). La calibration permet de définir le repère de la scène, en utilisant une des paires d'images, typiquement celle où la mire est posée à plat sur le bureau. Ainsi, le plan (X, Y) défini par la mire correspond au plan du bureau, et l'axe Z est perpendiculaire à la mire et au bureau (figure 2.3). L'axe Z correspond donc aussi, à quelques degrés près, aux axes optiques des caméras et donc à la dimension de profondeur du système stéréoscopique. La calibration permet donc de calculer des coordonnées 3D de points dans ce repère, en millimètres, ces points pouvant être le centre de la main ou les bouts des doigts.

2.3 LES CAMÉRAS

Les caméras vidéo utilisées pour ce projet sont des imageurs CMOS produit par STMICROELECTRONICS (figure 2.4). Ils permettent une acquisition aux formats CIF (352×288) ou VGA (640×480). Le modèle STV0676 est prévu pour être intégré dans des « webcams ». La génération la plus récente (VS6724) est destinée au marché grand public. Dans la suite de cette section, nous détaillons les caractéristiques du modèle STV0676, que nous utilisons pour l'acquisition de nos données.

Le flux vidéo est transmis à l'ordinateur par une connexion USB. Il peut être transmis au format YC_bC_r ou compressé au format MJPG par un coprocesseur. Par conséquent, il n'est pas possible de synchroniser l'acquisition entre les deux

(a) *image originale* (b) *image corrigée*

FIGURE 2.5 – *Correction de la distorsion radiale.*

caméras. Cette absence de synchronisation pose des problèmes dans le cas de la vision stéréoscopique avec deux caméras, puisque la main peut s'être déplacée entre l'acquisition des images des deux caméras. Nous étudions ce problème plus en détails dans le chapitre 6.

De plus, la connexion USB a un débit limité et monopolise une partie des ressources CPU de l'ordinateur. Cette limitation de l'USB implique que, lors de l'acquisition avec deux caméras simultanément, la fréquence d'acquisition est limitée de 15 à 20 images par seconde. L'utilisation de caméras USB 2 devrait permettre d'augmenter la fréquence d'acquisition.

Par ailleurs, puisque les caméras sont dans un champ proche de la main, nous utilisons des optiques avec un grand angle de vue (90°), ce qui provoque des distorsions dans les images. La distorsion est essentiellement radiale (cf. paragraphe B.1.1). Il est possible de la corriger dans l'image entière en utilisant les paramètres calculés pendant la calibration (figure 2.5). Toutefois, faire la correction sur l'image entière est assez coûteux en temps de calcul, alors que cela n'est pas forcément nécessaire. En effet, lorsque nous calculons les coordonnées 3D de points par triangulation, il est suffisant de corriger la distorsion uniquement pour les points 2D (avant la triangulation).

2.4 GESTES UTILISÉS

Nous avons vu dans la section 1.1 que la main peut produire une grande diversité de gestes. Toutefois, il est extrêmement difficile de reconnaître toutes les configurations possibles de la main à partir de sa projection dans une image. En effet, suivant l'orientation de la main par rapport à la caméra, certaines parties de la main peuvent être cachées. Il est nécessaire de considérer des sous-ensembles de gestes en fonction de notre application.

Dans nos travaux, nous nous intéressons à plusieurs catégories de gestes : dans le chapitre 5, nous cherchons à reconnaître un ensemble défini de postures, qui permettraient d'envoyer des commandes à l'ordinateur. Les postures sont réalisées face à la caméra et nous considérons le cas des transformations euclidiennes (translation, rotation, changement d'échelle). Dans le chapitre 6, nous considérons d'abord les gestes déictiques en 3D, pour le remplacement de la souris par la main. Le cas du geste de pointage est celui qui nécessite le moins

d'hypothèses du fait de sa simplicité. La seule condition est de pouvoir détecter le doigt, il suffit donc que celui-ci soit visible. Nous cherchons ensuite à estimer la posture de la main de manière plus générale, sans connaissance a priori sur les configurations possibles. Dans ce cas, nous supposons que la main est face à la caméra, afin de pouvoir détecter les doigts. Dans le cas contraire, si la main est vue de côté, il n'est pas possible de connaître la configuration exacte de la main.

2.5 DONNÉES DE TEST

Cette section présente les données utilisées dans cette thèse pour l'évaluation des résultats. Nous utilisons deux types de données :

- Un *ensemble de séquences vidéo stéréoscopiques*, utilisé pour le suivi 3D des doigts et de la main (chapitre 6).
- Deux *bases de données d'images* contenant des postures de la main, utilisées pour la reconnaissance de postures 2D (chapitre 5), afin d'évaluer les performances de reconnaissance des différents descripteurs de formes. Pour commencer, nous utilisons la base de référence de TRIESCH, qui fournit un bon point de départ, mais qui est assez pauvre en nombre d'images et en diversité de configurations. Ensuite, nous définissons notre propre vocabulaire de gestes, pour procéder à l'acquisition de notre base de données.

2.5.1 *Séquences vidéos stéréoscopiques*

Pour l'évaluation du suivi des doigts et de la main, nous utilisons un ensemble de séquences stéréoscopiques, dont l'acquisition a été réalisée avec deux caméras calibrées au préalable, dans les conditions présentées dans la section 2.2.

Plusieurs types de séquences ont été utilisées :

- 25 séquences avec 1, 2, 3 et 5 doigts réalisant une trajectoire circulaire ou carrée, par exemple « `2doigts_cercle_L_1.avi` » pour la séquence n° 1 avec la caméra gauche, et 2 doigts réalisant un cercle ;
- 5 séquences où le nombre de doigts et la posture de la main sont variables, par exemple « `divers_L_1.avi` » pour la séquence n° 1 avec la caméra gauche.

La séquence `divers_L_1.avi` est par ailleurs utilisée dans le chapitre 4 pour comparer les différentes méthodes de segmentation et de détection.

Le tableau 2.1 présente le détail des postures réalisées dans cette séquence.

2.5.2 *Base de gestes de* TRIESCH

La base de gestes de TRIESCH ET VON DER MALSBURG [126] est une base de référence, utilisée dans plusieurs travaux, et mise à disposition sur Internet [6]. Elle est constituée de 10 postures de la main (figure 2.6), réalisées par 24 personnes, et devant trois fonds différents (blanc, noir et complexe).

Nous utilisons les ensembles d'images avec les fonds noir et blanc, ce qui représente 479 images [7]. Les images avec le fond complexe ne sont pas utilisées car

6. *Jochen Triesch Hand Posture Database* : `http://www.idiap.ch/resources/gestures/`
7. il manque une image pour le geste « v ».

IMAGE	ACTION	POSTURE	IMAGE	ACTION	POSTURE
14	apparition	0	246	1 doigt	5
16	poing	6	312	transition	0
37	transition	0	314	5 doigts	11
38	1 doigt	5	416	transition	0
74	transition	0	422	2 doigts + p.	3
77	2 doigts	2	475	transition	0
118	transition	0	479	poing	6
119	2 doigts + p.	3	498	transition	0
190	transition	0	501	5 doigts	11
192	2 doigts	2	528	transition	0
220	transition	0	533	poing	6
224	2 doigts	2	541	disparition	-
242	transition	0	557	fin	-

TABLEAU 2.1 – *Détail de la séquence « `divers_L_1.avi` », avec les postures correspondant au vocabulaire de notre base (cf. paragraphe 2.5.3), p. signifie pouce.*

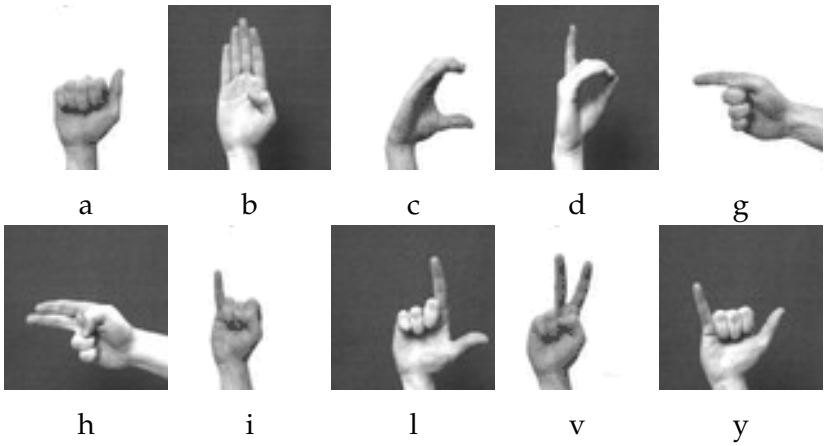


FIGURE 2.6 – *La base de gestes de* TRIESCH ET VON DER MALSBURG [126].

FIGURE 2.7 – *Exemple d'images de la base de* TRIESCH, *avec le geste « c ».*

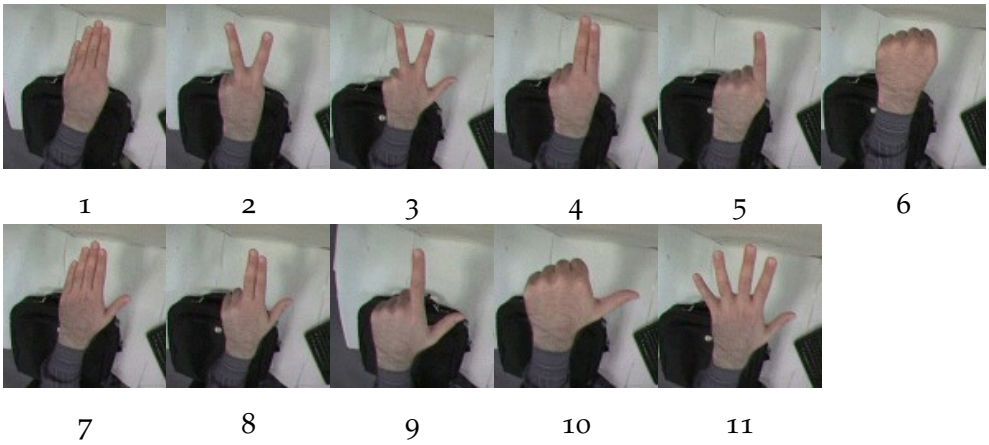


FIGURE 2.8 – *Les 11 gestes de notre base de données.*

notre objectif est ici de réduire l'influence de la segmentation sur les résultats de reconnaissance. Les images sont en niveaux de gris, au format PGM [8], et de taille 128×128. La variation de la forme des gestes en terme de taille, translation et rotation est très limitée. Par contre, la forme des mains des différents utilisateurs peut être très variable (figure 2.7).

2.5.3 *Notre base de gestes*

La base de gestes de TRIESCH est un bon point de départ, mais elle présente plusieurs limitations : le nombre d'images est faible, l'angle de prise de vue, la taille et l'orientation de la main sont toujours les mêmes, les images sont en niveaux de gris et ne contiennent que la main. Ainsi, pour réaliser des tests plus réalistes et plus proches de notre configuration, nous avons constitué notre propre base de gestes.

Notre base a été réalisée à partir de séquences vidéo monoscopiques. Les séquences vidéo ont ensuite été découpées en images, afin d'être traitées séparément.

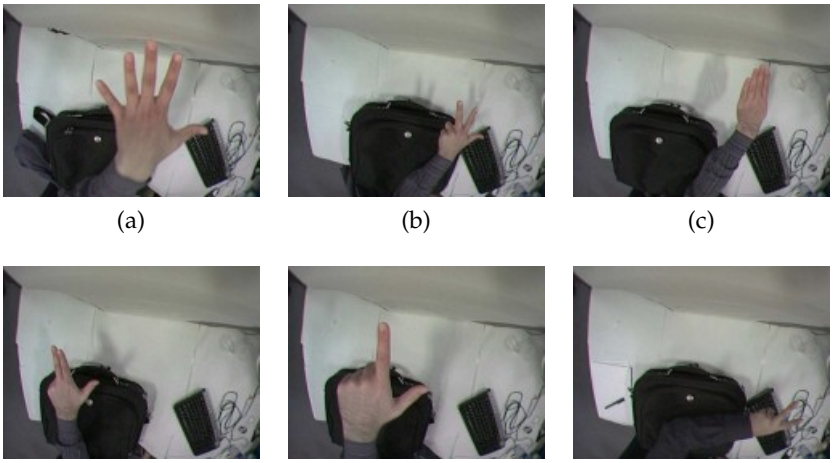
Les 11 gestes (figure 2.8) ont été choisis pour être facilement réalisables par un utilisateur quelconque. Ces gestes sont inspirés des 8 postures du *Langage Parlé Complété* (LPC) présentées par CAPLIER *et al.* [18]. Le LPC est un langage différent de la langue des signes, visant à faciliter la lecture sur les lèvres pour les personnes sourdes ou malentendantes. Toutefois, certains gestes ont été rajoutés afin de tester les performances de discrimination des descripteurs de formes. Certains gestes du LPC sont aussi très proches visuellement : c'est le cas des gestes 4 et 5, ainsi que des gestes 8 et 9.

18 personnes ont participé à la réalisation de cette base. La plupart d'entre elles ne sont pas familières avec la reconnaissance de gestes. En effet, tester la reconnaissance avec des personnes qui ne sont pas expertes dans le domaine constitue un aspect très important, afin d'évaluer si les gestes sont faciles à réaliser ou non, et comment leur réalisation peut varier d'une personne à l'autre.

L'acquisition des images s'est déroulée dans les conditions suivantes :

- environnement intérieur, avec un éclairage par des néons,
- les gestes sont réalisés au-dessus d'un bureau,

8. *Portable Gray Map*



(a) (b) (c)

(d) (e) (f)

FIGURE 2.9 – *Exemple d'images de notre base de données.*

- la caméra est une webcam positionnée au-dessus du bureau,
- la taille des images est 320×240,
- aucune supposition n'est faite sur le point d'entrée du bras dans la scène,
- les personnes portent des vêtements à manches longues afin de s'affranchir du problème de la détection du poignet pour séparer la main de l'avant-bras.

En effet, dans le cas où le bras des utilisateurs est nu, il faut séparer la main de l'avant-bras. Sinon, l'avant-bras est segmenté avec la main, et la forme utilisée pour la reconnaissance est alors trop différente des gestes du vocabulaire. Nous proposons une méthode pour détecter le poignet au paragraphe 4.3.2. Cependant, pour éviter que les résultats de la reconnaissance de postures ne soient faussés par une mauvaise détection du poignet, nous avons préféré demander aux personnes de porter un vêtement à manches longues pour l'acquisition des images de la base de données.

L'objectif de notre base de données de gestes est de pouvoir tester et comparer les performances des descripteurs de forme concernant les invariances en translation, rotation et changement d'échelle. C'est pourquoi nous avons demandé aux utilisateurs de bouger leur main dans tout l'espace de travail, en incluant la profondeur pour les changements d'échelle.

Nous avons ainsi obtenu environ 1 000 images par geste et par personne, soit un total d'environ 200 000 images. La figure 2.9 montre des exemples des images ainsi obtenues. La figure 2.9f illustre un cas difficile : la main est petite et représente peu de pixels par rapport à l'ensemble de l'image. Il est probable dans ce cas que la segmentation soit mauvaise et le contour peu représentatif de la forme.

Dans le chapitre suivant, nous présentons le domaine de l'interaction homme-machine gestuelle et nous réalisons un état de l'art de la reconnaissance visuelle de gestes et des problématiques associées.

3

INTERPRÉTATION DES GESTES DE LA MAIN

La main est à l'origine d'une grande variété de gestes. Différents dispositifs permettent d'interagir avec un ordinateur par l'intermédiaire de la main (souris, gants de données, écrans tactiles, etc.). Toutefois, ces périphériques souffrent de certaines limitations. Par ailleurs, les progrès scientifiques et techniques offrent de nouvelles possibilités d'interaction, plus naturelles et intuitives, basées sur le canal gestuel. Les applications sont nombreuses, de la réalité augmentée ou virtuelle, à la reconnaissance de la langue des signes, en passant par la commande de bras articulés ou encore la biométrie. Une des applications les plus développées consiste à rendre une surface interactive.

Différentes technologies ont été développées dans le but de reconnaître les gestes. Dans ce chapitre, nous présentons un état de l'art des approches basées sur la vision par ordinateur, dans le contexte des Interfaces Homme-Machine. L'interprétation visuelle des gestes offre l'interaction la plus naturelle et intuitive, mais elle est aussi la plus difficile à mettre en oeuvre. Au cours des dernières années, la reconnaissance visuelle de gestes a fait l'objet de nombreux travaux. Il est donc difficile de réaliser un état de l'art exhaustif du domaine. Nous tentons dans ce chapitre de donner une image représentative des différentes approches existantes. Des compléments de bibliographie plus spécifiques seront apportés dans les chapitres suivants.

Nous présentons les difficultés liées à la reconnaissance de gestes en vision par ordinateur, puis nous abordons la question de la modélisation des gestes. De manière générale, un système de reconnaissance de gestes peut se décomposer en plusieurs étapes [105] : modélisation, analyse et reconnaissance. On peut distinguer deux grandes familles de modèles : les modèles d'apparence et les modèles 3D. Par ailleurs, nous distinguons également deux sous-catégories : les gestes de pointage et les gestes dynamiques.

SOMMAIRE

3.1 VERS UNE INTERACTION HOMME-MACHINE GESTUELLE

Cette section présente les différents dispositifs permettant d'interagir avec un ordinateur. Nous présentons ensuite les applications et les nouvelles possibilités d'interaction basées sur la reconnaissance des gestes de la main.

3.1.1 *Dispositifs d'interaction*

La majorité des systèmes d'exploitation repose sur le paradigme WIMP, avec un dispositif de pointage, généralement la souris, qui permet d'interagir avec des éléments graphiques tels que des fenêtres, des icônes et des menus. Il existe aussi des interfaces *haptiques*, qui permettent un retour d'information à l'utilisateur, avec un retour de toucher ou un retour d'effort. La perception *tactilo-kinesthésique*, ou *haptique*, résulte de la stimulation de la peau par le contact avec des objets.

Cette section présente les différents dispositifs permettant une interaction avec l'ordinateur : les périphériques d'entrée (souris et autres dispositifs de pointage), les gants de données, les caméras vidéo et les écrans tactiles.

3.1.1.1 Périphériques d'entrée

Pour interagir avec un ordinateur, la souris s'est imposée comme le périphérique d'entrée indispensable. Il existe aussi la *boule de commande*, ou « *trackball* » (figure 3.1a). Le *pavé tactile* ou « *touchpad* » est un dispositif de pointage spécifique aux ordinateurs portables, permettant de remplacer la souris. Il s'agit d'une surface sensible de faible dimension, utilisant la capacité électrique.

Pour les jeux vidéo, le périphérique usuel est la manette ou « *joystick* ». Récemment, les constructeurs ont cherché à développer d'autres types d'interactions que les boutons, avec par exemple des vibrations pour le retour d'effort, ou des capteurs de mouvements. Ainsi, la `Wiimote`[1] (figure 3.1b) de NINTENDO a marqué un tournant dans les périphériques de jeux vidéo. Cette manette est équipée de capteurs qui lui permettent de se repérer dans l'espace et de retranscrire les mouvements de l'utilisateur à l'écran.

(a) (b)

FIGURE 3.1 – *Exemples de périphériques d'entrée : (a) le trackball* `TrackMan`[2] *de* LOGITECH, *et (b) la* `Wiimote` *de* NINTENDO.

1. http://wiiportal.nintendo-europe.com/1029.html
2. http://www.logitech.com/index.cfm/mice_pointers/trackballs/&cl=fr,fr

(a) (b)

FIGURE 3.2 – *Exemples de gants de données : (a) le* `5DT Data Glove 5 Ultra`*, et (b) le* `CyberGlove`*.*

3.1.1.2 Les gants de données

Par rapport à la richesse de l'information véhiculée par des gestes de la main, les possibilités de communication avec les ordinateurs sont réduites avec la souris et le clavier. Des dispositifs spécialisés pour une application sont apparus. Ainsi, pour l'acquisition de données en trois dimensions, des périphériques d'entrée 3D fournissent à l'ordinateur des informations sur la position de la main, voire sur sa configuration pour les plus évolués.

C'est le cas des *gants de données* (ou *gant électronique*, *gant numérique*), qui sont munis de capteurs fournissant la position de la main et les angles des articulations des doigts. Le `CyberGlove`[3] par exemple, qui peut être utilisé avec le système `Polhemus`[4], et qui fournit toutes les informations sur la configuration de la main, ou le `Dataglove`[5] également très utilisé (figure 3.2).

Les gants de données sont utilisés de longue date pour la reconnaissance de la langue des signes [12, 94, 125], car ils fournissent les positions précises et fiables des articulations de la main. Malheureusement, ces gants ont un coût élevé et sont encombrants, leur utilisation est contraignante pour l'utilisateur.

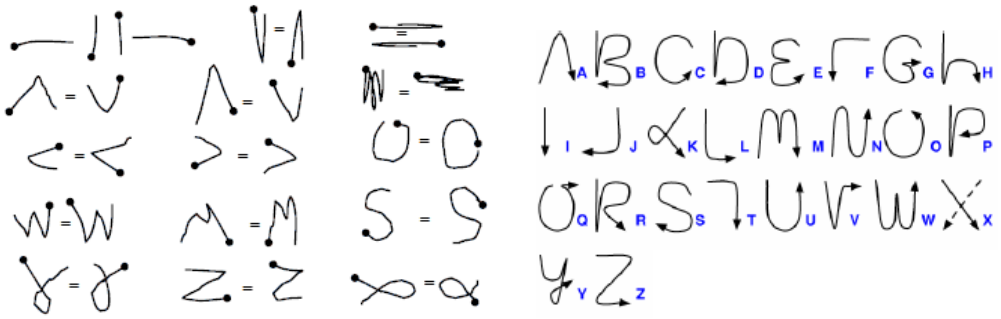
3.1.1.3 Écrans tactiles

Les écrans tactiles combinent à la fois entrée et sortie, avec la visualisation sur un écran et un dispositif de pointage pour interagir directement avec l'information affichée à l'écran. Cette technologie est utilisée pour des écrans de la taille d'un moniteur d'ordinateur, par exemple pour les guichets de billetterie automatique ou pour les « assistants personnels digitaux » (PDA[6]) avec éventuellement l'utilisation complémentaire d'un stylet.

Il existe différentes technologies pour les écrans tactiles : capacitive, résistive, infrarouge, à ondes de surface... De manière générale, les écrans tactiles souffrent de différents inconvénients : ils sont onéreux, limités en surface, sujet à l'usure et sensibles aux rayures (suivant la technologie employée).

Les écrans tactiles permettent la reconnaissance de gestes simples, appelés *gestes de dessins*. Par exemple, MERTZ *et al.* [95] utilisent les gestes de commande présentés sur la figure 3.3a. Il existe aussi des alphabets simplifiés pour les PDA (figure 3.3b), permettant la réalisation de lettres d'un seul trait.

3. IMMERSION, CyberGlove, `http://www.immersion.com/3d/products/cyber_glove.php`
4. IMMERSION, Polhemus, `http://www.immersion.com/3d/products/polhemus.php`
5. FIFTH DIMENSION TECHNOLOGIES, http://www.5dt.com
6. *Personal Digital Assistant*
7. iGesture : *A General Gesture Recognition Framework*, http://www.igesture.org



(a) *Gestes de commande utilisés par* Mertz *et al.* [95]*, avec un écran tactile.*

(b) *Alphabet simplifié* iGesture[7].

Figure 3.3 – *Exemples de gestes de dessins.*

3.1.1.4 Les caméras vidéos

Contrairement aux systèmes précédents, les caméras vidéos captent les mouvements de la main sans que l'utilisateur ne soit contraint de porter un équipement particulier, ou d'utiliser un périphérique dédié. Toutefois, pour certains systèmes, des marqueurs ou un gant coloré sont utilisés pour faciliter la détection des différentes parties de la main. La difficulté de cette approche est de mettre au point des traitements robustes pour interpréter le flux vidéo et extraire l'information utile de la grande quantité d'information disponible.

Par ailleurs, une caméra ne fournit qu'une information 2D. Pour obtenir des informations en 3D, il faut utiliser deux ou plusieurs caméras, ou une modélisation 3D. Par conséquent, les occultations sont un problème important, inhérent à la projection de l'espace 3D dans une image.

Avec l'augmentation de la puissance des ordinateurs de bureau et l'apparition de caméras bon marché, il est désormais possible de développer des systèmes de reconnaissance de gestes fonctionnant en temps réel. C'est à cette problématique que nous nous intéressons dans le cadre de cette thèse.

Les caractéristiques telles que le taux de rafraîchissement ou la résolution varient d'une caméra à l'autre. Des valeurs élevées de ces caractéristiques sont avantageuses pour avoir une mise à jour fréquente des images et un niveau de détail important. Toutefois, un compromis est nécessaire car des valeurs trop importantes augmentent la complexité et le temps de traitement des données. Par ailleurs, les caractéristiques du capteur (CCD ou CMOS) et de l'optique ont une influence primordiale sur la qualité de l'image obtenue et la sensibilité à l'éclairage.

3.1.2 *Applications et nouvelles possibilités d'interaction*

Cette section présente différentes applications, et notamment les surfaces interactives, ainsi que quelques applications récentes destinées au grand public.

3.1.2.1 Reconnaissance de la langue des signes

La reconnaissance de la langue des signes est une application naturelle de la reconnaissance de gestes. En effet, un système de traduction automatique des signes est particulièrement intéressant pour les sourds et muets. Braffort

[12] présente une étude détaillée de la Langue des Signes Française (LSF). Cette langue possède un vocabulaire et une syntaxe, et repose sur de nombreux paramètres :

- la *configuration* : pour représenter les objets ;
- le *mouvement* : pour représenter les actions ;
- l'*emplacement* : pour indiquer où sont effectuées les actions ;
- l'*orientation* : pour conjuguer certains verbes, ou préciser l'orientation des objets ;
- la *mimique faciale* : pour exprimer le mode du discours.

Ce bref aperçu révèle la richesse et la complexité de la LSF, et il en est de même pour les autres langues. C'est pourquoi la grande majorité des systèmes de reconnaissance de la langue des signes utilise des gants numériques (paragraphe 3.1.1.2), qui permettent d'obtenir des paramètres sur la configuration de la main plus facilement qu'avec la vision.

Dans les travaux consacrés à la reconnaissance de la langue des signes en vision par ordinateur (Ong et Ranganath [102]), le vocabulaire est généralement restreint à un sous-ensemble de gestes. La référence dans ce domaine est les travaux de Starner et Pentland [120][121], qui s'intéressent à la reconnaissance de la langue des signes américaine (ASL [8]), avec un vocabulaire de 40 mots. Ils obtiennent un excellent taux de reconnaissance de plus de 90%. La majorité des systèmes de reconnaissance, avec des gants numériques ou en vision par ordinateur, repose sur l'utilisation de *Modèles de Markov Cachés* (HMM [9], cf. section 3.6).

3.1.2.2 Réalité virtuelle

La réalité virtuelle consiste à plonger l'utilisateur dans un environnement d'images de synthèse [63, 114]. Cette immersion dans un environnement virtuel peut se faire grâce à un casque, ou dans une pièce dédiée munie de plusieurs écrans ou d'un vidéo-projecteur. L'utilisateur est donc en immersion complète dans un environnement dans lequel il peut interagir, notamment par le biais de gestes.

3.1.2.3 Réalité augmentée

La réalité augmentée mélange le monde physique et des informations virtuelles, en surimposant ces informations dans le champ de vision de l'utilisateur. Ce type de système est fondé sur une analyse par l'ordinateur du monde environnant l'utilisateur, au moyen d'un système de vision, de type caméra vidéo par exemple, ou de capteurs spécifiques. Des informations virtuelles sont alors projetées sur des éléments réels, par exemple par l'intermédiaire d'un vidéo-projecteur.

Une des applications principales de la reconnaissance de gestes en réalité augmentée est de rendre une surface interactive : une table, un tableau ou un bureau, sur laquelle des images sont projetées. L'utilisateur peut alors interagir avec des objets réels ou virtuels. Ces systèmes permettent aussi une interaction multi-utilisateurs. Ils sont présentés plus en détails dans la section suivante.

8. *American Sign Language*
9. *Hidden Markov Models*

FIGURE 3.4 – *Le système* `HandVu` *de* KOLSCH *et al.* [79].

Un autre exemple d'application est proposé par KOLSCH *et al.* [79], avec le système `HandVu` (figure 3.4), une interface gestuelle basée sur la vision. Ce système fonctionne avec un casque intégrant une caméra et un dispositif de visualisation ainsi qu'un microphone. Ce système démontre la faisabilité d'une interface gestuelle utilisant la vision.

3.1.2.4 Surfaces d'interaction

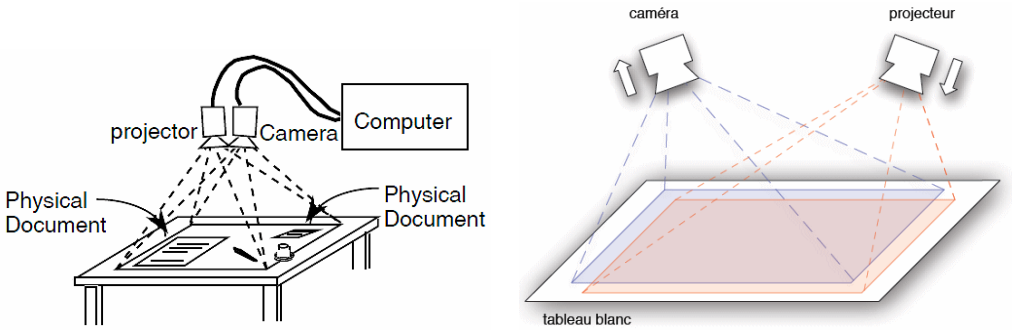
Le tableau blanc conventionnel est toujours très utilisé, pour donner des enseignements, noter les choses à faire, mettre ses idées au clair, ou comme support de réunion pour le travail collectif. Toutefois, il souffre de certaines limitations, notamment pour la gestion de l'espace et la réorganisation des données, ou l'absence de sauvegarde. Des solutions électroniques permettent de résoudre ces problèmes, en offrant une surface interactive. Une étude réalisée par LACHENAL [84] distingue les systèmes suivants :

- Les *surfaces à électronique intégrée*, ou *tableaux électroniques* : un dispositif matériel est intégré à la surface. Les tableaux peuvent être à projection arrière, tactiles ou fondés sur la technologie scanner.
- Les *surfaces à électronique externalisée* ou *tableaux augmentés* : un tableau blanc conventionnel est augmenté par des capteurs et des effecteurs, comme des instruments, par exemple un stylo avec un capteur de pression et qui émet des ultrasons, ou des caméras vidéo.
- Les *surfaces pour un usage collectif* : une surface pour le travail collectif, avec une gestion des utilisateurs.

Intéressons-nous en détail au cas de la caméra vidéo jouant le rôle de capteur. La caméra peut être utilisée de plusieurs façons, plus ou moins complexes. La plus simple est de l'utiliser comme un scanner, permettant de numériser le contenu de la surface observée, et donc de l'éditer, de l'imprimer, ou de l'envoyer par voie électronique. La caméra peut aussi servir à interpréter des commandes écrites sur le tableau, voire à suivre des instruments et plus particulièrement le doigt. C'est le cas des tableaux augmentés par la vision que nous présentons dans le paragraphe suivant.

LES TABLEAUX AUGMENTÉS PAR LA VISION

Le `DigitalDesk` (« *Bureau Digital* ») de WELLNER [134] est un des tous premiers prototypes de surface interactive pour la réalité augmentée. Dès 1991, ce système permet une interaction à deux doigts, comme sur un écran tactile actuel, mais sans écran. Une caméra et un projecteur sont disposés au-dessus d'un bureau



(a) (b)

FIGURE 3.5 – *Exemples de tableaux augmentés : (a) le* `DigitalDesk` *de* WELLNER [134] *et (b) la* `Table Magique` *de* BÉRARD [13].

(figure 3.5a), et permettent de faire des « copier-coller » avec les doigts, et de mélanger informations réelles et virtuelles. De nombreux systèmes ont ensuite été développés, s'inspirant de ce prototype.

L'`EnhancedDesk` (« *Bureau Augmenté* ») de SATO *et al.* [112][100] permet le suivi des doigts de plusieurs utilisateurs, et la reconnaissance de trajectoires 2D. Une caméra infrarouge est utilisée pour faciliter la segmentation.

La `Table Magique` (BÉRARD [13][14], HALL *et al.* [50]) [10] est une variante du `DigitalDesk` : c'est un espace de travail augmenté par l'apport de la vision par ordinateur. Le dispositif (figure 3.5b) est constitué d'un tableau blanc horizontal combiné avec une caméra numérique et un projecteur vidéo. Il permet de capturer des gestes et de projeter un retour d'information. Ce système permet de combiner la manipulation d'encre physique et d'encre virtuelle (projetée), en se fixant comme contrainte de maintenir l'utilisabilité des outils existants (feutres, brosse et doigts).

LES TECHNOLOGIES DÉDIÉES

Différentes technologies ont été développées ces dernières années, pour rendre une surface interactive. Ces technologies permettent d'étudier les possibilités de l'interaction gestuelle, les points intéressants, les limitations, et de voir ce qui peut être réalisable avec la vision par ordinateur.

HAN [51] a développé une technologie d'écran tactile reposant sur le renvoi de lumière diffusée [11], et qui permet une interaction multi-points. Un autre avantage est que cette technologie fonctionne pour de grandes surfaces. Une interface a ensuite été développée, et une vidéo de démonstration sur internet [12] permet de visualiser les possibilités d'interaction qu'offre le système.

La société SENSITIVE OBJECT [13] a pour sa part développé une technologie d'interaction gestuelle basée sur la reconnaissance des ondes acoustiques. Une des applications est de remplacer le clavier physique par un clavier dessiné sur le bureau.

Ces exemples laissent envisager de nouvelles possibilités d'interaction avec la main. Les technologies dédiées aux surfaces interactives permettent de résoudre

10. http://iihm.imag.fr/demos/magicboard/
11. *frustrated total internal reflection* (FTIR)
12. http://www.perceptivepixel.com
13. http://www.sensitiveobject.fr

FIGURE 3.6 – *Le système* `Surface` *de* MICROSOFT.

certains problèmes liés à la vision par ordinateur. Toutefois, elles sont limitées à un type d'application précis, alors que la vision permet une souplesse et une adaptation plus importantes.

3.1.2.5 Applications grand public

Ces dernières années, plusieurs applications grand public ont émergé. Ces nouvelles possibilités d'interaction sont illustrées par le film *Minority report* [14], où l'on peut voir un acteur manipuler des données numériques avec les mains, sur un écran géant. Les technologies commercialisées visent à capter les mouvements de l'utilisateur pour le placer au centre d'un jeu vidéo, ou à permettre la manipulation de données numériques avec une interface interactive, éventuellement par plusieurs personnes simultanément.

Par exemple, le système `EyeToy` [15] développé par SONY pour la console de jeu PLAYSTATION2 utilise une petite caméra USB pour capter les mouvements du corps humain, et pour modéliser la tête de l'utilisateur en 3D.

Le système `Surface` [16] (figure 3.6) de MICROSOFT utilise des caméras pour détecter des gestes de la main, ou des objets. Le résultat est affiché sur une surface par rétro-projection. Ainsi, les utilisateurs peuvent utiliser leurs mains pour interagir avec leurs données numériques. Le système peut être utilisé simultanément par plusieurs personnes, qui peuvent se regrouper autour de la surface interactive. Il est aussi possible de placer des objets physiques sur la surface, par exemple pour transférer des données avec des appareils numériques.

3.2 INTERPRÉTATION VISUELLE DES GESTES DE LA MAIN

Au cours des quinze dernières années, l'interprétation visuelle des gestes de la main a été le sujet de nombreux travaux de recherche. Avec les progrès scientifiques et techniques, il est aujourd'hui possible d'envisager une interaction homme-machine plus naturelle et intuitive, basée sur la reconnaissance de gestes en vision par ordinateur. D'une part, la reconnaissance de gestes permet de dépasser les limitations des périphériques d'entrée classiques (paragraphe 3.1.1) en offrant de nouvelles possibilités d'interaction. D'autre part, la vision par

14. http://fr.wikipedia.org/wiki/Minority_Report
15. http://fr.playstation.com/ps2/hardware/accessories/
16. http://www.microsoft.com/surface/

ordinateur permet de se passer de périphériques encombrants, coûteux et contraignants, tels que les gants numériques. Une ou plusieurs caméras suffisent.

Nous allons voir dans les sections suivantes que de nombreuses approches ont été étudiées pour résoudre le problème de l'interprétation visuelle des gestes. Une grande partie de ces approches a été développée en se concentrant sur un aspect particulier du geste, tel que le suivi de la main, l'estimation de la posture, ou la classification de gestes. Ces études ont souvent été menées dans le cadre d'une application particulière, telle que la commande à distance d'une télévision, la navigation dans un environnement virtuel, ou la reconnaissance de la langue des signes. Ces limitations proviennent du fait qu'il est difficile d'aborder la reconnaissance de gestes dans sa globalité, à cause de la variété des applications et des contraintes importantes liées à ce type de système.

Jusqu'à récemment, la majeure partie des travaux s'est concentrée sur les gestes statiques, ou postures, et sur les gestes de pointage. Ces deux aspects du gestes sont en effet les plus simples à reconnaître, et ils constituent logiquement la première brique d'un système plus complet de reconnaissance de gestes. Plus récemment, l'intérêt s'est porté sur les gestes dynamiques, qui consistent en un changement de la configuration ou de la position de la main, ou les deux simultanément. Ce dernier cas est donc le plus général.

3.2.1 *Difficultés liées à la vision par ordinateur*

Contrairement aux gants numériques, qui permettent d'obtenir directement des informations sur la position et la configuration de la main, la conception d'un système de vision par ordinateur nécessite la prise en compte d'un certain nombre de difficultés :

ÉCLAIRAGE DE LA SCÈNE : les variations de l'éclairage de la scène ont un impact important sur les algorithmes de segmentation et d'extraction de caractéristiques, en provoquant des variations de luminosité dans l'image.

OMBRES : la main projette une ombre dans la scène, même si la luminosité est contrôlée. Les ombres sont parfois segmentées avec la main, suivant la méthode utilisée, ce qui rend le résultat de segmentation difficilement exploitable.

OBJETS DE LA SCÈNE : la scène peut comporter des objets mobiles tels que le clavier, la souris ou des documents. L'algorithme de segmentation de la main doit pouvoir s'adapter à des changements du fond de la scène et distinguer la main des autres objets.

CAMOUFLAGE : lorsque les caractéristiques de l'objet sont proches de celles du fond, celui-ci est difficilement détectable. Dans le cas de la main, ce problème se produit par exemple lorsque le bureau a une couleur de teinte chair.

OCCULTATIONS : les occultations se produisent lorsqu'un objet est masqué par un autre, ce qui peut arriver lorsqu'on utilise les deux mains. Dans le cas de la main, il existe un problème supplémentaire avec les « auto-occultations » : la main étant un objet complexe et déformable, il arrive fréquemment que certaines parties, telles que les doigts, soient cachées par d'autres.

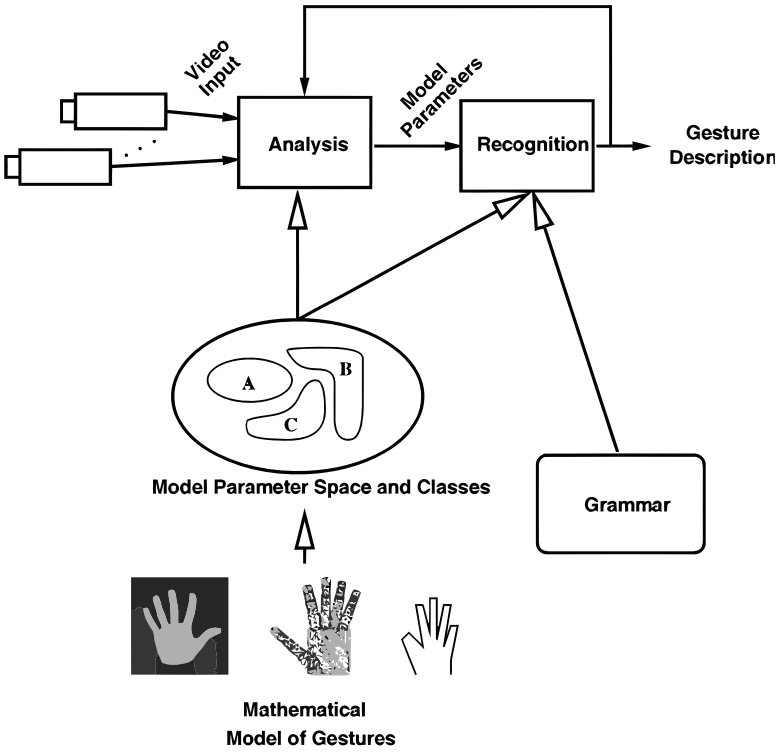


FIGURE 3.7 – *Représentation d'un système de reconnaissance de gestes (source :* PAVLOVIC *et al.* [105]*).*

DISPOSITION DES CAMÉRAS : la distance entre la caméra et les objets joue un rôle important, de même que l'angle de vue des caméras (90° dans notre cas) qui peut causer des déformations géométriques dans l'image.

Pour résoudre ces difficultés, différentes hypothèses sont émises en fonction de l'application visée, sur l'éclairage, sur le type d'objets présents dans la scène ou sur le type de gestes à reconnaître. Par exemple, pour segmenter la main de façon robuste aux variations de luminosité, certains systèmes utilisent un fond noir ou un gant de couleur (IWAI *et al.* [68]).

3.2.2 *Schéma général d'un système de reconnaissance de gestes*

D'une manière générale, un système de reconnaissance de gestes peut se décomposer en trois étapes (PAVLOVIC *et al.* [105], figure 3.7) : *modélisation*, *analyse* et *reconnaissance*.

3.2.2.1 Modélisation

La première étape consiste à choisir un modèle de geste, en considérant à la fois les caractéristiques spatiales et temporelles. La complexité du modèle dépend de l'application visée. Pour PAVLOVIC *et al.* [105], le geste peut être vu comme « un processus stochastique dans l'espace paramétrique des gestes sur un intervalle de temps donné » (section 1.1). Deux réalisations d'un même geste ne donnent pas le même vecteur de paramètres.

Le geste est un processus dynamique qui peut être découpé en trois phases (KENDON [74]) :
- *préparation* (à partir d'une position de repos),
- *noyau* (posture, trajectoire),
- *rétractation* (retour à la position de repos).

Un problème majeur pour la reconnaissance de gestes est de distinguer le début et la fin du geste. Un autre problème est de différencier un geste d'un mouvement non-intentionnel.

Un autre aspect important est la modélisation spatiale du geste. Les modèles peuvent être classés en deux grandes catégories :

MODÈLES D'APPARENCE : cette approche consiste à interpréter le geste directement à partir de l'apparence de la main dans les images, en le comparant avec un ensemble de gestes modèles. Les modèles d'apparence peuvent être basés sur les images, les moments géométriques, les positions des bouts des doigts, le contour, les vecteurs propres des images, ou encore un modèle 2D déformable représentant la forme moyenne du geste. Ces méthodes sont développées dans la section 3.4.
Cette approche permet généralement d'obtenir des traitements en temps réel. Par contre, elle impose des contraintes plus ou moins importantes sur l'environnement et sur les gestes. Son champ d'application est donc plus restreint que l'approche par modèle 3D.

MODÈLES 3D : l'autre approche possible consiste à utiliser un modèle 3D de la main, qui peut être *volumique* ou *squelettique*. L'intérêt de ces modèles est d'offrir une modélisation très élaborée de la main, en étant ajustés à la morphologie de la main, et prenant en compte les contraintes entre les articulations et les doigts. Les modèles volumiques permettent de représenter visuellement la main en 3D, par le biais de cylindres, sphères ou quadriques. Ils sont utilisés pour des approches dites « analyse par la synthèse » [17], qui consistent à faire varier les paramètres du modèle pour que celui-ci corresponde visuellement aux images. Les différentes méthodes sont détaillées dans la section 3.5.
Les modèles 3D peuvent être utilisés pour reconnaître les ensembles de gestes les plus riches. Toutefois, les inconvénients de cette approche sont la complexité des calculs, ce qui la rend peu adaptée pour le temps réel, et la difficulté d'estimer les paramètres par la vision.

3.2.2.2 Analyse

L'étape d'analyse consiste à calculer les paramètres du modèle à partir de caractéristiques extraites des images. Ces paramètres décrivent la posture ou la trajectoire de la main. Ils dépendent donc du modèle choisi. On peut distinguer deux étapes :
- extraction de caractéristiques à partir des images : localisation de la main, détection des bouts des doigts, signatures calculées à partir du contour de la main, etc. ;
- estimation des paramètres du modèle. Pour les modèles 3D par exemple, il s'agit d'estimer des paramètres angulaires (articulations) et linéaires (dimen-

17. *analysis-by-synthesis*

sions des phalanges). On distingue généralement une étape d'initialisation et une étape de mise à jour.

3.2.2.3 Reconnaissance

L'étape de reconnaissance consiste à classer les paramètres afin d'interpréter le geste. L'espace des paramètres est généralement partitionné lors d'une procédure d'apprentissage à partir d'exemples : un ensemble d'images d'apprentissage permet de calculer les paramètres des gestes modèles. Une métrique est ensuite utilisée pour classer le geste. Différents classifieurs peuvent être utilisés (boosting, réseaux de neurones, etc.). Pour les gestes dynamiques, les *Modèles de Markov Cachés* (HMM) sont les plus utilisés.

Dans la suite de ce chapitre, nous présentons les nombreux travaux dédiés à la reconnaissance de gestes. Cette étude est divisée en plusieurs sections consacrées aux différentes approches : les gestes de pointage, les modèles d'apparence, les modèles 3D et les gestes dynamiques.

3.3 GESTES DE POINTAGE

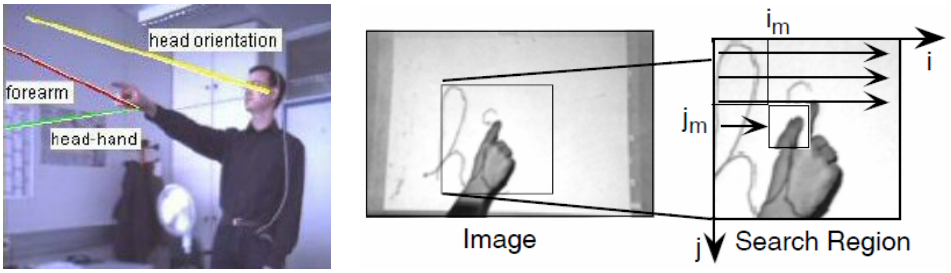
*cf. § 1.1 p. 2

Les *gestes de pointage** ont suscité de nombreux travaux, du fait de leur simplicité, et de leur application naturelle pour pointer et remplacer la souris. Ils peuvent aussi être utilisés pour dessiner des trajectoires qui sont ensuite reconnues. Les approches présentées dans cette section sont spécifiques aux gestes de pointage, mais d'autres travaux plus complets, présentés par la suite, traitent aussi ces gestes.

Il existe plusieurs approches pour déterminer la direction de pointage du doigt (figure 3.8a) : il est possible d'utiliser la position du doigt, sa direction, la droite entre l'œil et le doigt, ou encore la direction du bras. Ces estimations de position et de direction reposent généralement sur la vision stéréoscopique, mais il est également possible d'utiliser des informations géométriques (Wu *et al.* [136]).

Hung *et al.* [64] utilisent un système stéréoscopique calibré pour détecter la position pointée par un doigt. La détection de la main repose sur l'utilisation du seuillage de Otsu [103] avec un fond plus sombre que la main. Le doigt est séparé de la main par une opération morphologique et le bout de doigt est détecté comme étant l'extrémité du doigt. Le bout du doigt est suivi avec une recherche locale. Deux modes de pointage sont comparés : l'orientation 3D du doigt et la direction déterminée par l'œil et le bout du doigt. La deuxième solution s'avère plus précise et adaptée à un espace de travail plus vaste.

Jojic *et al.* [69] s'intéressent à la détection de gestes de pointage en temps réel. Leur approche est basée sur le calcul d'une carte de disparité, en vision stéréoscopique, pour détecter la personne par rapport au fond. L'estimation de la disparité est moins sensible aux variations de luminosité mais comporte d'autres facteurs d'incertitude (son estimation est peu fiable pour des zones uniformes). Ensuite, une modélisation statistique du corps humain permet de détecter le bras. La direction de pointage est déterminée par la composante principale de la région du bras.



(a) (b)

FIGURE 3.8 – *Exemple de gestes de pointage : (a) les différents modes de pointage : droite de vue « tête-main », direction du bras, orientation de la tête (source :* NICKEL *et al.* [99]*), et (b) détection du bout du doigt par corrélation (source :* CROWLEY *et al.* [31]*).*

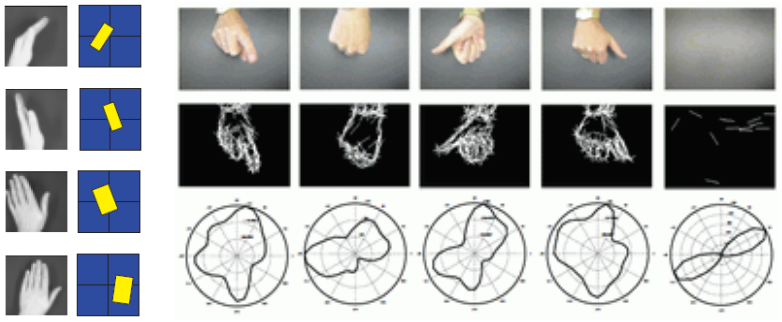
NICKEL *et al.* [99] combinent l'information de profondeur (carte de disparité) avec l'information de couleur de la peau, pour localiser les positions 3D de la tête et des mains. Les gestes de pointage sont décomposés en trois phases, chacune étant modélisée par HMM : *début*, la main se déplace pour pointer la cible ; *milieu*, la main pointe et reste fixe ; *fin*, la main se rétracte. Ils montrent aussi que les utilisateurs préfèrent intuitivement utiliser la droite de vue entre la tête et la main pour pointer (figure 3.8a).

WU *et al.* [136] utilisent une caméra à plusieurs mètres de l'utilisateur. Ils détectent les zones de couleur de la peau, identifient le bras et trouvent le bout du doigt avec la courbure du contour dont le calcul est simplifié en produit scalaire. La détection de la position du coude et de l'épaule permet d'obtenir une trajectoire en 3D, qui est ensuite lissée.

La mesure de corrélation est une méthode très employée pour la détection et le suivi. Le principe est de mémoriser l'apparence de l'objet avec une imagette, aussi appelée *motif*. La localisation du motif dans une nouvelle image s'effectue par un parcours de l'image, en comparant des blocs de l'image avec le motif, par une mesure de corrélation. La position de l'objet est alors le maximum de la fonction de corrélation. L'inconvénient de cette méthode est qu'elle suppose que l'apparence de l'objet ne change pas ou peu, ce qui n'est valable que si les déplacements de l'objet s'effectuent uniquement par des translations dans le plan de l'image. Mais si le motif à localiser change d'orientation ou de taille, le motif de référence ne correspond plus au motif à suivre.

Dans le système `DigitalDesk`, CROWLEY *et al.* [31] utilisent une caméra placée au-dessus d'un bureau pour effectuer un suivi du doigt par corrélation avec un modèle de bout de doigt (figure 3.8b). Le masque de référence est mis à jour pour prendre en compte les changements d'orientation du doigt. Pour rendre la mesure de corrélation plus robuste aux changements d'échelle et d'orientation, MARTIN [92] a proposé d'utiliser un ensemble de motifs à différentes orientations et échelles.

OKA *et al.* [100] utilisent la corrélation avec un cercle pour détecter les bouts de doigts dans des images binaires obtenues avec une caméra infra-rouge. FREEMAN ET WEISSMAN [44][42] utilisent aussi une corrélation normalisée pour contrôler une télévision par les mouvements de la main.



(a) (b)

FIGURE 3.9 – *Exemple de caractéristiques extraites des images (source :* FREEMAN *et al.* [42]*) : (a) Moments géométriques, (b) Histogramme d'orientation.*

3.4 MODÈLES D'APPARENCE

Nous avons présenté la modélisation par apparence dans le paragraphe 3.2.2 : cette approche consiste à utiliser des caractéristiques extraites des images pour reconnaître un geste parmi un ensemble prédéfini. Un apprentissage à partir d'exemples est généralement utilisé, afin d'apprendre les caractéristiques des gestes modèles. Les modèles d'apparence sont plus utilisés car ils sont moins complexes que les modèles 3D. Ils permettent de faire de la reconnaissance de gestes en temps réel, mais de façon moins générale que les modèles 3D.

Nous distinguons dans la suite les approches utilisant une seule caméra de celles basées sur la vision stéréoscopique. Il existe de nombreuses méthodes pour modéliser l'apparence de la main :

- *caractéristiques extraites des images :* silhouette, contour, moments géométriques, valeurs propres ;
- *positions des bouts des doigts :* cette approche suppose, sous certaines contraintes, que la connaissance des positions du bout des doigts par rapport à la paume est suffisante pour différencier un nombre fini de gestes ;
- *vecteurs propres :* l'ACP [18] permet de calculer la forme moyenne et de représenter les images avec peu de coefficients ;
- *méthodes statistiques :* histogrammes, modèles de distribution de points.

3.4.1 *Mono-caméra*

FREEMAN *et al.* [42] passent en revue différentes méthodes pour l'interprétation visuelle des gestes. Les moments du premier ordre donnent le centre de la région de la main, les moments du deuxième ordre permettent de calculer l'orientation de la main (figure 3.9a). Les histogrammes d'orientation permettent une représentation de la main peu sensible à l'éclairage et à la taille mais dépendante de l'orientation (figure 3.9b).

ATHITSOS ET SCLAROFF [4] utilisent un modèle 3D de la main pour générer des images synthétiques de postures de la main, à partir de 26 configurations de base (figure 3.10). Chacune de ces configurations est projetée avec 86 angles de vues, et pour chaque vue 48 images sont créées (car les mesures de similarité ne sont pas invariantes en rotation). Chaque image est classifiée avec une recherche

18. Analyse en Composantes Principales

FIGURE 3.10 – *Les 26 configurations de la main utilisées par* ATHITSOS ET SCLAROFF [4].

hiérarchique de l'image la plus similaire dans la base de données. Quatre mesures de similarité sont combinées : une distance du chanfrein entre les contours, un histogramme de l'orientation des contours, une mise en correspondance des bouts de doigts, et une mise en correspondance basée sur les moments centrés et les moments de HU. Ils améliorent ensuite la méthode (ATHITSOS ET SCLAROFF [5]) en la rendant plus robuste aux problèmes de segmentation, en utilisant deux nouvelles mesures de similarité : une méthode du chanfrein améliorée, et une mise en correspondance probabiliste des lignes.

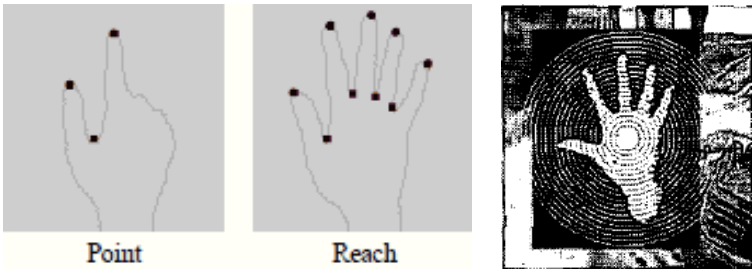
HOLDEN ET OWENS [58] détectent la main grâce à la couleur de la peau par ACP [19] et avec une distance de MAHALANOBIS. Le centre de la région de la main est suivi par un algorithme de *Condensation*. Des caractéristiques de la forme de la main sont extraites de cette région en la convertissant en coordonnées polaires.

3.4.2 *Vision stéréoscopique*

SEGEN ET KUMAR [114] utilisent deux caméras calibrées pour suivre la main. Des points de contours caractéristiques (bouts des doigts et creux inter-doigts) sont détectés avec la courbure du contour. Ces points permettent de classer quatre gestes de la main : *pointer*, *atteindre*, *cliquer* et *repos* (figure 3.11a). Ils déterminent la direction pointée par le doigt en 3D, avec 10 degrés de liberté. Leur système fonctionne à 60 Hz, et a été appliqué à un jeu vidéo, un simulateur de vol et à la commande d'un bras de robot.

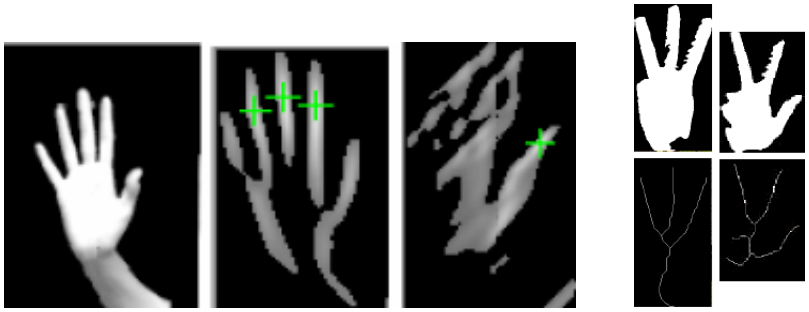
YIN ET XIE [140] reconnaissent des postures 2D de la main en analysant des caractéristiques topologiques. La main est segmentée par une approche colorimétrique, basée sur l'espace couleur L*a*b*. Un réseau de neurones RCE [20] permet d'identifier les pixels dont la couleur est celle de la peau, après un apprentissage. Les bouts de doigts sont détectés comme étant les points de contours les plus éloignés du centre de gravité de la main (figure 3.11b). Les postures 2D de la main sont alors caractérisées par le nombre de doigts et leur position. Ils étendent leur approche à la 3D, avec une caméra stéréoscopique non calibrée.

19. Analyse en Composantes Principales
20. *Restricted Coulomb Energy*



(a) (b)

FIGURE 3.11 – *Exemple de caractéristiques extraites des images : (a) détection des bouts de doigts et des vallées avec la courbure (source :* SEGEN ET KUMAR [114]*), et (b) détection des bouts de doigts avec la distance au centre de gravité (source :* YIN ET XIE [140]*).*



(a) (b)

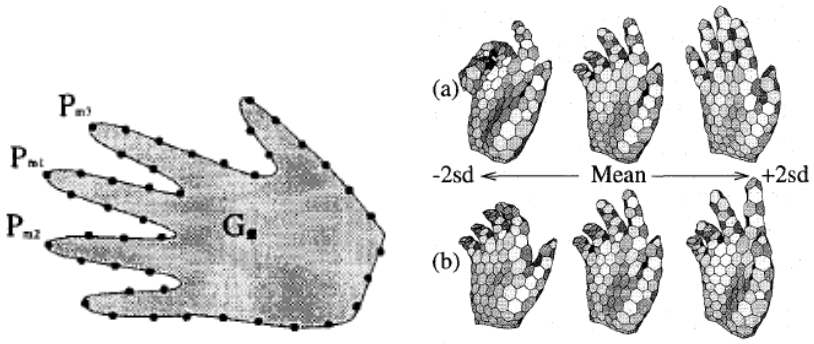
FIGURE 3.12 – *Exemple de caractéristiques extraites des images (source :* MACLEAN *et al.* [90]*) : (a) détection des doigts, et (b) squelettes.*

MACLEAN *et al.* [90] effectuent une segmentation basée sur la couleur de la peau, puis ils vérifient si la région segmentée est une main avec un filtrage détectant les doigts, en supposant qu'ils sont verticaux, et que le pouce est orienté à 45°. La reconnaissance de gestes est basée sur une squelettisation de la main, et un comptage du nombre de doigts (figure 3.12).

ABE *et al.* [1] ont mis au point une interface 3D pour manipuler des objets virtuels avec la main. Ils utilisent deux caméras avec des points de vues très différents (une caméra en haut et une à droite du bureau). Il existe deux modes différents : le pointage avec le calcul de la position 3D du doigt, et les commandes avec quinze postures, associées à des actions. Un fond noir leur permet de détecter la main facilement. Le poignet est détecté en utilisant une carte de distance, et un rectangle intérieur à la paume est calculé. La reconnaissance de postures se fait avec des fenêtres pour chaque doigt, positionnées à partir de la paume, et dans lesquelles les doigts sont détectés ou non.

3.4.3 *Analyse en composantes principales*

L'Analyse en Composantes Principales (ACP) est utilisée pour extraire un sous-espace optimal d'une distribution de points. Cette technique est très utilisée pour la reconnaissance de visages, où l'on parle d'« *eigenface* » (TURK ET PENTLAND [127]). L'ACP est aussi utilisée pour calculer la forme moyenne de la main, avec les modèles de distribution de points (cf. paragraphe 3.4.4).



(a) (b)

FIGURE 3.13 – *Exemple de modèles déformables : (a) modèle de distribution de points (source :* SHIMADA *et al.* [117]*), et (b) modèle déformable* 3D *(source :* HEAP ET HOGG [57]*).*

MOGHADDAM ET PENTLAND [96] ont appliqué leur méthode d'estimation de l'espace des vecteurs propres sur des images de contour de la main. Ils proposent un modèle d'« *eigenhand* » permettant de localiser la main et de reconnaître des gestes.

MARTIN [92] compare l'ACP avec le discriminant linéaire de FISHER, pour la reconnaissance de huit postures de la main. Il propose ensuite d'utiliser une « distance à l'espace propre » pour classer les gestes. CUI ET WENG [32] préfèrent aussi l'analyse discriminante linéaire, fournissant la « *caractéristique la plus discriminante* » [21], à l'ACP. En effet, l'analyse discriminante linéaire optimise la discrimination, alors que l'ACP optimise la reconstruction.

3.4.4 *Modèles déformables*

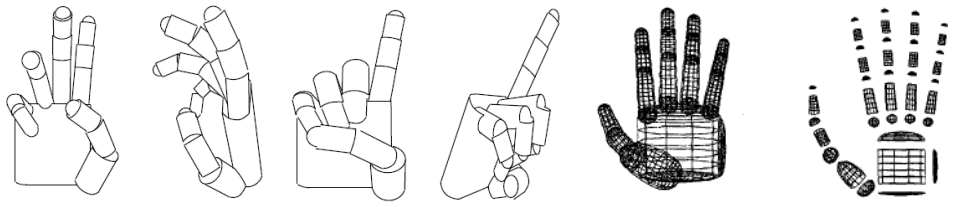
Le « modèle de distribution de points » [22] a été introduit par COOTES *et al.* [28] [29]. Il s'agit d'un modèle 2D déformable, qui consiste en un ensemble de points répartis sur le contour de la main et décrivant la forme moyenne. Le modèle de distribution de points est décrit par des paramètres obtenus par ACP [23] sur un ensemble d'apprentissage. Il est ensuite déformé pour correspondre à la main de l'utilisateur.

Ce modèle est aussi utilisé par AHMAD *et al.* [3] pour suivre la main et reconnaître cinq gestes, et par HEAP ET SAMARIA [56] pour les « *smart snakes* », technique utilisant les contours actifs pour le suivi d'objets déformables.

HEAP ET HOGG [57] ont étendu cette approche à la 3D (figure 3.13b) en effectuant l'apprentissage du modèle avec des images à résonance magnétique (IRM) et en l'appliquant au suivi de la main avec une caméra.

SHIMADA *et al.* [117] estiment la posture 3D de la main avec une seule caméra. Ils comparent le contour 2D d'une main avec une base de données de modèles d'apparences possibles de la main, contenant 16 000 images de contours générées à partir d'un modèle 3D. La forme de la main est décrite en plaçant un ensemble de N points sur le contour, avec un intervalle régulier (figure 3.13a).

21. *Most Discriminant Features* (MDF)
22. *Point Distribution Model* (PDM)
23. Analyse en Composantes Principales



(a) (b)

FIGURE 3.14 – *Exemple de modèles* 3D *de la main : (a) modèle* 3D *de la main de* REHG ET KANADE [111], *et (b) modèle* 3D *de la main de* STENGER *et al.* [124].

3.5 MODÈLES 3D

L'approche par modèle 3D consiste à reconstruire la main en trois dimensions. Le modèle est mis en correspondance avec une ou plusieurs images, afin d'estimer ses paramètres (dimensions et angles des articulations). Ce type de modèle permet de reconnaître un ensemble de gestes plus riche que les modèles d'apparence. Il permet de prendre en compte la morphologie de la main, ainsi que les contraintes entre les articulations et entre les doigts (section 6.4). L'intérêt est de connaître la configuration exacte de la main à chaque instant, afin de résoudre les problèmes d'occultations entre les doigts. Cette connaissance de la posture exacte de la main est intéressante pour les gestes manipulatifs, mais elle n'est pas utile pour les gestes communicatifs.

Il existe différents types de modèles 3D : les modèles *volumiques* et les modèles *squelettiques*. Les modèles volumiques sont constitués de cylindres, sphères ou quadriques, ce qui permet de décrire l'apparence visuelle de la main en 3D. L'étape d'apprentissage n'est pas nécessaire, même si elle est parfois utilisée pour obtenir des informations supplémentaires. L'estimation des paramètres par la vision est une tâche complexe, qui implique généralement une étape d'initialisation et une étape de mise à jour. L'initialisation est parfois utilisée afin d'ajuster le modèle à la morphologie de la main de l'utilisateur, ce qui se fait avec la main ouverte, doigts écartés. Les modèles squelettiques sont basés sur des caractéristiques morphologiques et biomécaniques, afin de représenter les phalanges et les articulations.

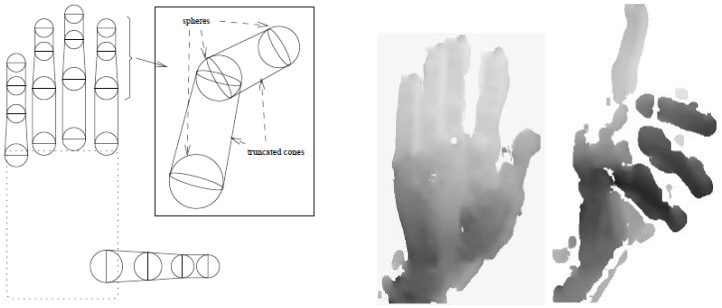
Il existe deux types d'approches pour faire correspondre un modèle 3D avec les images :

- Recalage du modèle dans les images, qui consiste à faire varier les paramètres du modèle pour que celui-ci corresponde visuellement aux images, en fonction de critères tels que le taux de recouvrement des silhouettes [81, 118] ou la distance entre les contours [47, 118].
- Extraction de caractéristiques (par exemple les bouts des doigts), et estimation du modèle.

On peut aussi distinguer deux types de méthodes : celles qui sont capables de fonctionner en temps réel au prix de concessions sur certains points, notamment le nombre de gestes reconnus et la précision ; et celles qui cherchent à retrouver la position exacte de la main sans se préoccuper du temps de calcul.

REHG ET KANADE [111] ont introduit l'utilisation d'un modèle 3D articulé de la main avec le système `DigitEyes`. Leur modèle à 27 degrés de liberté [24]

24. 4 degrés de liberté pour chaque doigt, 5 pour le pouce et 6 pour la paume.



(a) (b)

FIGURE 3.15 – *Le modèle 3D de la main de* DELAMARRE ET FAUGERAS [37] *: (a) le modèle 3D avec des cônes et sphères, et (b) exemple de carte de disparité utilisée pour le recalage.*

(figure 3.14a) est devenu une référence dans ce domaine. L'état de la main est estimé en minimisant l'erreur entre les positions détectées des bouts de doigts et les projections du modèle 3D dans les images. Ce système a été utilisé pour une application de souris 3D avec une caméra, et pour l'estimation de la posture de la main en vision stéréoscopique.

LEE ET KUNII [86] utilisent un modèle 3D squelettique, pour la reconnaissance de 16 symboles de la langue des signes américaines (ASL [25]). Leur modèle est basé sur la détection de points caractéristiques (extrémités des doigts de la main, position du poignet, ainsi qu'un point supplémentaire sur la paume), détectés grâce à un gant coloré. Leur algorithme comporte cinq types de contraintes pour les articulations, et le processus de recalage se divise en deux phases : d'abord le poignet, puis la paume et les doigts.

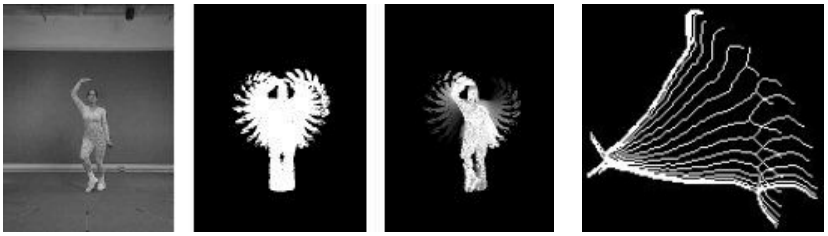
DELAMARRE ET FAUGERAS [37] utilisent un modèle 3D articulé à 27 degrés de libertés, basé sur celui de REHG. Le modèle 3D est mis en correspondance avec la carte de disparité issue de la reconstruction 3D de la main (figure 3.15). Le recalage est réalisé avec des forces appliquées au modèle.

OUHADDI ET HORAIN [104] recalent un modèle 3D articulé de la main avec une caméra. La main est détectée avec la teinte de la peau et les doigts identifiés par filtrage morphologique. Pour le recalage, deux fonctionnelles sont comparées, le taux de non-recouvrement des silhouettes et la distance entre les contours occultants du modèle projeté dans l'image ; ainsi que deux méthodes d'optimisation, la descente du simplexe et la méthode de POWELL. L'intégration de contraintes biomécaniques dans la procédure de recalage permet de réduire considérablement l'espace de recherche. Cette méthode leur a fourni des résultats assez concluants, mais on notera que certaines configurations ne peuvent être retrouvées, telles que le mouvement de flexion d'un doigt, de face.

STENGER *et al.* [124] utilisent un modèle 3D construit à l'aide de quadriques (figure 3.14b), pour générer des vues 2D qu'ils comparent avec les images. Ils utilisent une ou deux caméras et retrouvent l'orientation de la main 3D avec un filtre de KALMAN, qui minimise l'erreur géométrique entre les vues générées et les contours extraits des images.

D'autres approches existent et sont référencées notamment par OUHADDI ET HORAIN [104], GAVRILA [46] et DELAMARRE [36].

25. *American Sign Language*



(a) (b)

FIGURE 3.16 – *Gestes dynamiques : (a) MEI et MHI (source :* BOBICK ET DAVIS [9]*), et (b) signature dynamique d'un geste avec la superposition des squelettes (source :* IONESCU *et al.* [66]*)*

3.6 GESTES DYNAMIQUES

Un geste dynamique correspond à une variation temporelle de la forme et de la position de la main. La première difficulté est de localiser temporellement la réalisation d'un geste, c'est-à-dire de déterminer le début et la fin du geste. Un geste se déroule en trois étapes : une phase de préparation, le geste, et une phase de rétractation (paragraphe 3.2.2).

Une difficulté importante provient de la variation de la durée de réalisation d'un même geste. Il est donc nécessaire d'effectuer une normalisation temporelle de la durée des observations. Le *recalage dynamique* (DTW [26]) permet de comparer deux séquences temporelles de durées différentes, en étirant ou en réduisant leur longueur, ce qui suppose que le début et la fin du geste sont bien déterminés. DARRELL ET PENTLAND [34] utilisent cette méthode : les gestes sont modélisés par des scores de corrélation avec un ensemble de modèles, qui sont accumulés pour former une signature. Le recalage dynamique permet de comparer les signatures.

BOBICK ET DAVIS [9] utilisent des modèles temporels pour la reconnaissance du mouvement humain : l'« *image de l'énergie du mouvement* » (MEI [27]), et l'« *image de l'historique du mouvement* » (MHI [28]). Ces images sont formées par l'accumulation du mouvement de chaque pixel sur une fenêtre temporelle (figure 3.16a). Les images obtenues sont décrites avec les invariants de HU, et les gestes sont classifiés en utilisant la distance de MAHALANOBIS.

IONESCU *et al.* [66] proposent une méthode de reconnaissance de gestes dynamiques basée sur les squelettes. Des signatures statiques de début et de fin des gestes sont calculées avec un histogramme des orientations du gradient. La signature dynamique est obtenue par superposition des squelettes des images de la séquence (figure 3.16b).

ZHU *et al.* [144] segmentent la main avec la couleur, associée à la détection de mouvement. La représentation spatio-temporelle d'un geste se fait avec l'estimation du mouvement, basée sur un modèle paramétrique, et une description de la forme de la main avec les moments géométriques. Après une normalisation temporelle avec une méthode de ré-échantillonnage linéaire, la reconnaissance est effectuée avec une distance avec les modèles des gestes appris

26. *Dynamic Time Warping*
27. *Motion Energy Image*
28. *Motion History Image*

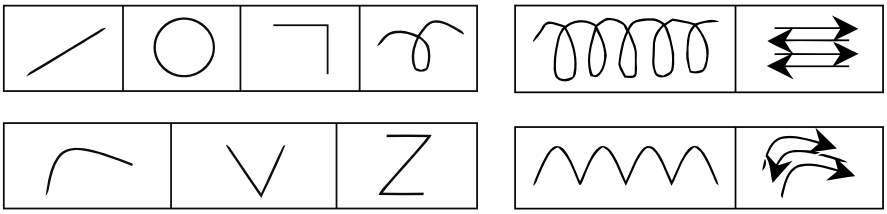


FIGURE 3.17 – *Trajectoires 3D utilisées par* KONG ET RANGANATH [80] *: (a) non périodiques, et (b) périodiques*

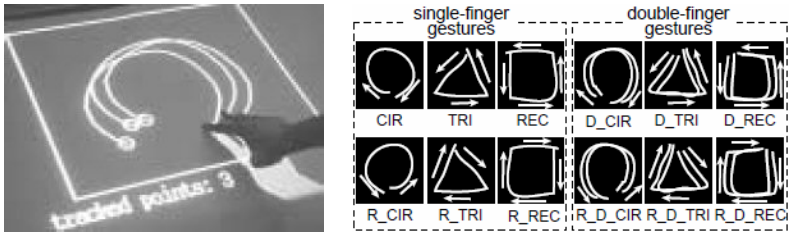
auparavant. Dans leur application, 12 gestes sont utilisés pour naviguer dans une vue panoramique.

KONG ET RANGANATH [80] utilisent une approche hiérarchique pour reconnaître des trajectoires 3D [29], périodiques ou non (figure 3.17). La détection de la périodicité est basée sur une analyse de FOURIER. Les trajectoires sont ensuite reconnues avec une variante de l'ACP [30].

Les *Modèles de Markov Cachés* (HMM) sont utilisés avec succès depuis longtemps dans le domaine de la reconnaissance de la parole. Par analogie, ils ont été utilisés pour la reconnaissance de gestes et l'interprétation de la langue des signes, d'abord avec des gants numériques (BRAFFORT [12]), puis en vision par ordinateur où différents modèles ont été développés. Parmi les premiers travaux dans ce domaine, STARNER ET PENTLAND [120][121] utilisent les HMM pour la reconnaissance de 40 signes issus de la langue des signes américaine (ASL), avec une seule caméra. Les caractéristiques utilisées sont le centre de la main et la boîte englobante elliptique, obtenue avec les axes principaux. MARCEL *et al.* [91] proposent une approche hybride entre les HMM et les réseaux de neurones, appelée « Input-Output Hidden Markov Models », pour reconnaître quatre gestes, en utilisant le centre de gravité de la main. WILSON ET BOBICK [135] proposent une forme paramétrique des HMM, pour estimer la direction du mouvement dans un geste de pointage. VOGLER ET METAXAS [129][130] proposent les « Parallel HMM » pour modéliser séparément les mains gauche et droite, et reconnaître 53 gestes de la langue des signes américaine, de façon continue.

SATO *et al.* [112][100] réalisent un suivi de la main et des bouts de doigts, en deux dimensions, pour le système `EnhancedDesk`. Une caméra infra-rouge facilite la détection des mains, puis chaque bout de doigt est détecté par corrélation avec un cercle, et suivi avec un filtre de KALMAN. Le pouce est détecté pour différencier un mode « *manipulation* » d'un mode « *geste symbolique* ». La reconnaissance des gestes symboliques repose sur l'utilisation des HMM, avec 12 gestes différents (figure 3.18). De la même façon, MARTIN ET DURAND [93] utilisent les HMM pour la reconnaissance d'écriture en 2D, avec des lettres tirées d'un alphabet pour les PDA.

29. *Signing Exact English* (SEE)
30. *Vector Quantization Principal Component Analysis* (VQPCA)



(a) (b)

FIGURE 3.18 – *Le système* `EnhancedDesk` *(source :* OKA *et al.* [100]*) : (a) suivi de plusieurs bouts de doigt, et (b) trajectoires reconnues par HMM.*

3.7 RÉSUMÉ

Dans ce chapitre, nous avons d'abord présenté les différents dispositifs d'interaction et les nombreuses applications de la reconnaissance de gestes, avec notamment la reconnaissance de la langue des signes et la réalité augmentée. Des applications grand public ont vu le jour, telle que le système `Surface` de MICROSOFT. Ces applications utilisent des technologies variées, telles que la diffusion de la lumière, les ondes acoustiques ou la vision.

Nous avons ensuite explicité les contraintes liées à la reconnaissance de gestes en vision par ordinateur, ce qui en fait un domaine de recherche très actif, avec de nombreuses approches différentes. Nous avons vu que la reconnaissance de gestes peut se décomposer en trois étapes principales : modélisation, analyse et reconnaissance.

Les deux grands types de modèles que sont les modèles d'apparence et les modèles 3D ont été détaillés, avec leurs avantages et inconvénients. Les modèles d'apparences sont plus simples et adaptés pour les systèmes en temps réel, mais ils sont restreints à certaines applications. Ils sont bien adaptés à la reconnaissance d'un ensemble de gestes défini, notamment pour les gestes communicatifs. Les modèles 3D visent à retrouver la posture exacte de la main et permettent de reconnaître un ensemble de gestes plus riche que les modèles d'apparence, mais les temps de calculs peuvent être très importants. Ils sont plus adaptés aux gestes manipulatifs.

Par ailleurs, nous avons souligné que les gestes de pointage ont suscité de nombreux travaux, avec des applications de type « souris 3D ». Enfin, nous avons évoqué les gestes dynamiques, qui représentent la forme la plus générale du geste, avec à la fois des variations de position et de configuration. L'application généralement visée est la reconnaissance d'un sous-ensemble de gestes de la langue des signes.

Du fait de la littérature abondante dans le domaine, de nombreux travaux n'ont pu être détaillés ni même mentionnés dans ce chapitre. Pour approfondir cet état de l'art, nous renvoyons le lecteur aux articles de PAVLOVIC *et al.* [105], KOHLER ET SCHROTER [76], LAVIOLA [85], WU ET HUANG [137], ONG ET RANGANATH [102]. Dans les chapitres suivants, nous nous intéressons aux gestes de pointage, pour le suivi d'un doigt, aux approches par apparence, pour le suivi de la main et la reconnaissance de postures, et aux modèles 3D pour la visualisation du suivi 3D.

4

DÉTECTION ET CARACTÉRISATION MORPHOLOGIQUE DE LA MAIN

Dans ce chapitre, nous étudions les méthodes de détection et de segmentation de la main dans les images d'un flux vidéo, avec une caméra. L'objectif est d'imposer le moins de contraintes possibles à l'utilisateur, et d'utiliser des méthodes adaptées au contexte de notre application. En effet, de telles méthodes reposent généralement sur un certain nombre d'hypothèses sur l'environnement, intérieur ou extérieur, sur l'éclairage, ou sur la couleur du fond. Dans *notre configuration**, il s'agit d'un environnement intérieur, permettant d'éviter le problème des fortes variations de luminosité, mais il reste le problème des ombres.

**cf.* § 2.2 *p.* 10

Nous présentons trois approches, et nous discutons de leur adéquation au cas de la segmentation de la main :

SEUILLAGE DE OTSU : méthode de seuillage pour les images en niveaux de gris, avec un seuil calculé à partir de l'histogramme.

DIFFÉRENCE D'IMAGES : la soustraction entre deux images met en valeur leurs différences. Elle peut se faire avec une image de référence du fond (partie statique de la scène), pour détecter des objets, ou entre deux images successives, pour détecter les objets en mouvement.

DÉTECTION DE LA COULEUR DE PEAU : la couleur de la peau a une distribution caractéristique dans certains espaces colorimétriques, et cette propriété peut être utilisée pour segmenter les régions de couleur de peau, comme les mains ou le visage. Les méthodes existantes reposent généralement sur une étape d'apprentissage, pour calculer un modèle de couleur de la peau, à partir d'images où la main est segmentée.
Nous proposons une méthode pour rendre l'apprentissage automatique, et pour mettre à jour le modèle de couleur de la peau. Celui-ci est basé sur des histogrammes dans l'espace YC_bC_r.

Ensuite, nous nous intéressons à l'extraction de caractéristiques de position basée sur la morphologie de la main, à partir de l'image binaire et du contour. Les caractéristiques extraites sont le centre de la main, le poignet, et les bouts des doigts. Elles sont utilisées par la suite pour le suivi 3D des doigts et de la main (chapitre 6).

SOMMAIRE

4.1 INTRODUCTION

La détection de la main dans les images d'un flux vidéo est une problématique importante, commune à la plupart des systèmes de reconnaissance de gestes. Cette première étape est primordiale, car elle conditionne les résultats de la suite des traitements. La détection de la main dans une séquence vidéo est un sujet de recherche à part entière [20, 78, 101, 143]. Il existe de nombreuses méthodes, plus ou moins performantes suivant les suppositions faites sur la scène, et suivant l'environnement (intérieur, extérieur, condition d'illumination...).

De plus, nous souhaitons non seulement détecter la main (c.-à-d. savoir si elle est présente dans l'image), mais aussi la segmenter (c.-à-d. savoir quels pixels de l'image appartiennent à la main) afin d'obtenir son contour, avec la meilleure précision possible, pour en extraire des informations sur sa forme et sa position.

Par ailleurs, il faut prendre en compte la contrainte temps réel, car la segmentation de la main n'est que la première étape d'un système de reconnaissance de gestes, qui en compte plusieurs autres. Il faut donc trouver un compromis entre les performances de la segmentation et la rapidité d'exécution.

Dans une approche de vision par apparence, la main est représentée par un ensemble de caractéristiques de forme et de position. Celles-ci peuvent être spatiales et temporelles, et être utilisées pour le suivi et la reconnaissance de gestes. Ces informations sont extraites à partir des images en couleurs ou en niveaux de gris, des images binaires ou du contour de la main, obtenus par une étape de segmentation.

Dans ce chapitre, nous nous intéressons plus particulièrement aux caractéristiques de localisation du centre de la main, du poignet et des bouts des doigts. Cependant, d'autres valeurs peuvent être utiles, comme la surface ou l'orientation de la main.

4.2 SEGMENTATION DE LA MAIN

Afin de simplifier la segmentation, pour se concentrer sur le suivi et la reconnaissance, il est possible de restreindre le problème en utilisant un fond de couleur uniforme, en demandant aux utilisateurs de porter un gant de couleur (IWAI *et al.* [68]) ou en positionnant des marqueurs colorés sur le bout de leurs doigts (DAVIS ET SHAH [35]). Mais ces artifices ne sont pas satisfaisants pour une application grand public, car trop contraignants pour l'utilisateur. Une autre solution consiste à utiliser une caméra infra-rouge (OKA *et al.* [100]), qui permet de détecter la main en utilisant la différence de chaleur par rapport au fond.

CHEN *et al.* [20] proposent de combiner plusieurs méthodes (différence temporelle d'images, couleur de la peau, et détection de contour) avec un opérateur logique ET sur les images binaires. Ceci leur permet de localiser la main, ils réalisent ensuite une soustraction du fond pour obtenir précisément la forme de la main. Cette combinaison de méthodes permet de rendre la détection plus robuste, mais au prix d'un temps de calcul élevé.

L'information de profondeur, obtenue par la vision stéréoscopique, a été utilisée pour détecter la main par DELAMARRE ET FAUGERAS [37] et JOJIC *et al.* [69]. En effet, à partir d'une paire d'images stéréoscopiques, il est possible de calculer une *carte de disparité** dans laquelle la main se distingue du fond, les valeurs de la carte de disparité étant inversement proportionnelles à la distance

*cf. § B.3 p. 128

à la caméra. Toutefois, le calcul d'une carte de disparité est relativement coûteux en temps de calcul, et n'est pas très fiable pour des surfaces uniformes. NICKEL *et al.* [99] ont combiné cette information de profondeur avec celle de couleur.

Il est également possible de ne pas chercher à segmenter précisément la main, mais de simplement détecter sa présence dans l'image, et sa position le cas échéant. La mesure de corrélation est très utilisée pour détecter la main, mais aussi les bouts des doigts (cf. section 3.3). D'autres méthodes de détection peuvent être utilisées, comme le détecteur de VIOLA et JONES (KOLSCH ET TURK [78]) ou le « *Boosted Classifier Tree* » (ONG ET BOWDEN [101]).

4.2.1 *Seuillage de* OTSU

Le seuillage d'une image en niveaux de gris I consiste à associer la valeur 0 (noir) à tous les pixels dont la valeur est inférieure au seuil α et la valeur 1 (blanc) à tous les autres. On obtient une image seuillée S, aussi appelée *masque binaire* ou *silhouette* :

$$\forall (x,y),\ S(x,y) = \begin{cases} 1, & \text{si } I(x,y) \geq \alpha \\ 0, & \text{sinon} \end{cases} \tag{4.1}$$

L'inconvénient de cette méthode est qu'il faut déterminer le seuil manuellement, de façon empirique, et qu'un seuil donné est adapté à des conditions d'illumination bien précises. Or, il est très difficile de trouver un seuil qui donne de bons résultats dans toutes les situations.

Il existe des méthodes plus robustes où le seuil est calculé automatiquement en fonction de l'image. C'est le cas de la méthode de OTSU [103], pour laquelle le seuil est calculé à partir de l'histogramme des niveaux de gris de l'image. Cette méthode est adaptée à des images dans lesquelles les deux classes sont bien définies. L'histogramme $h(i)$ de l'image est séparé en deux classes, dont on calcule les moyennes μ_1 et μ_2, et variances σ_1^2 et σ_2^2, en fonction du seuil T :

$$\mu_1 = \frac{1}{T}\sum_{i=0}^{T-1} h(i) \qquad \sigma_1^2 = \frac{1}{T}\sum_{i=0}^{T-1} (h(i)-\mu_1)^2 \tag{4.2}$$

$$\mu_2 = \frac{1}{256-T}\sum_{i=T}^{255} h(i) \qquad \sigma_1^2 = \frac{1}{256-T}\sum_{i=T}^{255} (h(i)-\mu_2)^2 \tag{4.3}$$

Si l'image comporte N pixels, on a les probabilités des deux classes :

$$p_1 = \frac{1}{N}\sum_{i=0}^{T-1} h(i) \qquad p_2 = \frac{1}{N}\sum_{i=T}^{255} h(i) = 1 - p_1 \tag{4.4}$$

Le seuil optimal est celui qui minimise la variance intra-classe σ_w^2 :

$$\sigma_w^2 = p_1\sigma_1^2 + p_2\sigma_2^2 \tag{4.5}$$

La figure 4.1 montre le résultat du seuillage avec la méthode de OTSU et avec un seuil fixe. On constate que le résultat n'est pas parfait, qu'il y a des « trous » dans le masque. Ce problème sera résolu par les post-traitements présentés au paragraphe 4.2.4.

(a) *image originale* (b) *résultat avec un seuil à 128* (c) *résultat avec la méthode de* Otsu (d) *histogramme de l'image*

Figure 4.1 – *Exemple de segmentation par seuillage, sur une image de la base de gestes de* Triesch *(cf. paragraphe 2.5.2).*

4.2.2 *Différence d'images*

La détection d'objets par différence d'images consiste à soustraire une image par une autre, pixel à pixel, ce qui suppose que la caméra soit fixe afin qu'un pixel de l'image représente toujours le même lieu de l'espace au cours du temps. Cette approche se situe au niveau pixel, ce qui signifie qu'elle ne prend pas en compte les relations qui existent entre des pixels voisins.

La valeur des pixels est supposée stable dans le temps. Cependant, les variations de luminosité de la scène peuvent faire varier cette valeur. Ces méthodes sont donc très sensibles aux variations de luminosité, aux ombres, et aux changements du fond. Il faut également prendre en considération le bruit d'acquisition de la caméra.

Il existe deux types de méthodes pour la différence d'images :
- La *différence d'images successives*, qui détecte les objets en mouvement.
- La *soustraction du fond*, qui détecte les objets n'appartenant pas au fond.

4.2.2.1 Différence d'images successives

Avec I_t l'image courante et I_{t-1} l'image précédente, l'*image de différence* D s'obtient par :

$$D(x, y, t) = |I(x, y, t) - I(x, y, t-1)| \tag{4.6}$$

Cette image de différence est ensuite seuillée pour mettre en évidence les zones en mouvement. Cela concerne à la fois les changements positifs (un objet est apparu) et les changements négatifs (un objet est disparu). Comme le montre la figure 4.2b, il est difficile d'extraire le contour de l'objet à partir de cette image de différence.

4.2.2.2 Soustraction du fond

La soustraction du fond permet de détecter les objets au premier plan. Le fond correspond à la partie statique de la scène, qui ne change pas au cours du temps. La méthode consiste à soustraire une image de référence $I_{\text{réf}}$, correspondant à la scène sans objet, à l'image courante I_t :

$$D(x, y, t) = |I(x, y, t) - I_{\text{réf}}(x, y)| \tag{4.7}$$

L'image de référence est prise à l'initialisation du système, en supposant qu'il n'y a pas d'objet dans la scène. Elle peut aussi être calculée en moyennant les

(a) *image originale* (b) *différence entre I_t et I_{t-1} seuillée* (c) *différence avec l'image de référence* (d) *image (c) seuillée*

FIGURE 4.2 – *Exemple de segmentation par différence d'images.*

N premières images de la séquence. La soustraction peut être effectuée avec l'image en niveaux de gris, ou en couleur sur chacune des trois composantes RGB. L'image différence est seuillée (par exemple avec la méthode de OTSU) pour obtenir un masque binaire.

Le problème de cette méthode est sa sensibilité aux variations de luminosité. En effet, si la valeur des pixels correspondant au fond varie de façon trop importante, ces pixels sont détectés comme des objets lors du seuillage. La méthode proposée par BERTOLINO *et al.* [6] permet de détecter un changement global de luminosité, lent ou rapide, faisant varier le fond de façon importante (lumière allumée ou éteinte). De la même façon, les ombres des objets sont détectées, car elles provoquent une variation significative de la valeur des pixels. La figure 4.2d illustre la difficulté de segmenter précisément la main avec cette méthode, principalement à cause des ombres.

RÉACTUALISATION DU FOND

Pour améliorer la robustesse de cette méthode face au problème des variations de luminosité, il est possible de réactualiser l'image de référence par une moyenne temporelle avec l'image courante, avec α une constante de temps déterminant la vitesse de réactualisation :

$$I_{\text{réf}}(x,y,t) = (1-\alpha)\, I_{\text{réf}}(x,y,t-1) + \alpha\, I(x,y,t) \tag{4.8}$$

De cette façon, l'évolution du fond est prise en compte dans l'image de référence. Cette moyenne revient à faire un oubli exponentiel, avec $1/\alpha$ la constante de temps du processus d'oubli. Toutefois, cette approche suppose que le fond représente la plus grande partie de la scène, et que les objets restent en mouvement. Sinon les objets sont progressivement intégrés au fond, en fonction de la valeur du paramètre α.

Cette méthode de réactualisation par moyennage améliore la segmentation, mais reste trop globale. Le choix du paramètre α est délicat : s'il est trop grand, on risque d'intégrer des objets au fond ; s'il est trop petit, l'image de référence ne s'adapte pas assez rapidement aux variations de luminosité.

Il est possible d'améliorer la soustraction du fond en modélisant les pixels par un mélange de gaussiennes, qui sont réactualisées au fil du temps en fonction des résultats de segmentation. Cette approche a été proposée par STAUFFER ET GRIMSON [122][123] et reprise dans de nombreux travaux. Elle a été étudiée par CONSEIL [23] et est présentée plus en détail dans l'annexe A.

SUPPRESSION DES OMBRES

La détection des ombres est un problème important pour les méthodes de soustraction du fond. En effet, l'ombre de l'objet est elle-même détectée, et il est alors impossible de retrouver le contour exact. Il existe différentes méthodes pour déterminer si un pixel détecté correspond à un pixel du fond, ombragé ou non (PRATI *et al.* [109]).

Nous présentons une méthode non-paramétrique, basée sur les travaux de HORPRASERT *et al.* [60], HARITAOGLU *et al.* [53] et KAEWTRAKULPONG ET BOWDEN [72], et qui possède l'avantage d'être simple à mettre en oeuvre. Cette méthode se base sur le fait qu'un pixel du fond ombragé (ou inversement, illuminé par le soleil) a approximativement la même couleur mais une luminosité plus faible (ou inversement, plus forte) que le pixel dans son état « normal ».

Pour chaque pixel i, notons $I_i = [I_R(i), I_G(i), I_B(i)]$ sa valeur courante, et $E_i = [E_R(i), E_G(i), E_B(i)]$ sa valeur du fond. La méthode repose sur le calcul de la distorsion de luminosité α_i (minimisant la fonction ϕ) et la distorsion chromatique CD_i, entre la valeur courante et la valeur du fond :

$$\phi(\alpha_i) = (I_i - \alpha_i E_i)^2 \tag{4.9}$$

$$CD_i = ||I_i - \alpha_i E_i|| \tag{4.10}$$

En appliquant des seuils sur ces deux mesures, il est alors possible de détecter les pixels correspondant au fond ombragé. Toutefois, ces seuils sont déterminés de manière empirique, et peuvent varier d'une séquence vidéo à une autre. Cette solution n'est donc pas satisfaisante, car elle nécessite une supervisation de l'utilisateur.

4.2.3 *Détection de la couleur de peau*

L'analyse de la couleur de la peau est très utilisée pour la détection du visage et des mains. En effet, JONES ET REHG [70] ont montré que la couleur de la peau présente une distribution caractéristique dans certains espaces colorimétriques, et que cette propriété peut être utilisée pour segmenter les régions de couleur de peau. Il existe deux aspects importants dans ce type de segmentation : le choix d'un espace de couleur et le choix de la méthode pour classer les pixels de peau.

Nous utilisons l'espace YC_bC_r, qui consiste en une composante de luminance (Y) et deux de chrominance (C_b et C_r). La transformation est linéaire avec l'espace RGB. Il existe de nombreux autres espaces colorimétriques, les plus utilisés étant RGB, HSV et YC_bC_r. PHUNG *et al.* [107] ont comparé les performances de ces espaces et ils constatent que les résultats sont très similaires, quelque soit l'espace couleur. Ainsi, le choix d'un espace colorimétrique doit se faire en fonction du format des images et d'éventuels pré-traitements. Dans notre cas, le flux vidéo fournit par les caméras utilise un espace YC_bC_r échantillonné, ce qui motive notre choix.

Il existe différentes méthodes pour classer les pixels de couleur de la peau. La plupart des approches sont basées sur un modèle de couleur de peau prédéterminé, obtenu avec un apprentissage « hors-ligne ». Un ensemble d'images est sélectionné pour l'apprentissage, dans lesquelles la main est segmentée, parfois de façon manuelle. Cette tâche peut donc vite devenir fastidieuse.

On distingue deux grands types de modèles pour la classification :

PARAMÉTRIQUE : la distribution de la couleur de peau est modélisée par un mélange de gaussiennes. Une étape d'apprentissage permet de calculer les paramètres des gaussiennes, et ainsi de calculer la probabilité qu'un pixel donné soit un pixel de peau. Cette méthode suppose que la distribution de couleur de peau puisse être modélisée par une gaussienne, ce qui n'est pas forcément évident. Le mélange de gaussiennes peut par exemple être estimé avec l'algorithme EM [1] (ZHU *et al.* [143]). Il peut aussi être mis à jour pour adapter le modèle aux changements de luminosité.

NON-PARAMÉTRIQUE : la distribution de la couleur de peau est modélisée par un histogramme, 2D ou 3D. Ceci a l'avantage de ne pas faire de supposition sur le type de distribution. On peut ensuite calculer la probabilité qu'un pixel donné soit un pixel de peau. Le temps de calcul nécessaire est plus faible que pour les méthodes paramétriques, car il suffit de prendre la valeur dans l'histogramme, mais l'espace de stockage nécessaire est plus important, puisqu'il faut stocker l'histogramme en mémoire au lieu de quelques paramètres pour les gaussiennes.

PHUNG *et al.* [107] ont réalisé une comparaison des différentes méthodes, et ils montrent que la méthode avec les histogrammes et la classification bayésienne est parmi les plus performantes. Ils montrent également que les performances sont moins bonnes lorsque la chrominance seule est utilisée pour la classification. La prise en compte de la composante de luminance dans l'histogramme permet d'être plus robuste aux variations de luminosité. Les méthodes basées sur un mélange de gaussiennes sont plus gourmandes en temps de calcul, alors que leurs performances ne sont pas meilleures.

Dans les paragraphes suivants, nous présentons d'abord une méthode simple, basée sur un seuillage des composantes C_b et C_r. Puis nous nous intéressons à une méthode utilisant la modélisation de la couleur de peau par des histogrammes.

4.2.3.1 Seuillage C_bC_r

La méthode la plus simple pour segmenter la peau est le seuillage. Il suffit de définir explicitement les bornes d'une région d'un espace de couleur, correspondant à la couleur de la peau. Par exemple, les bornes suivantes ont été proposées par CHAI ET NGAN [19] pour l'espace couleur YC_bC_r :

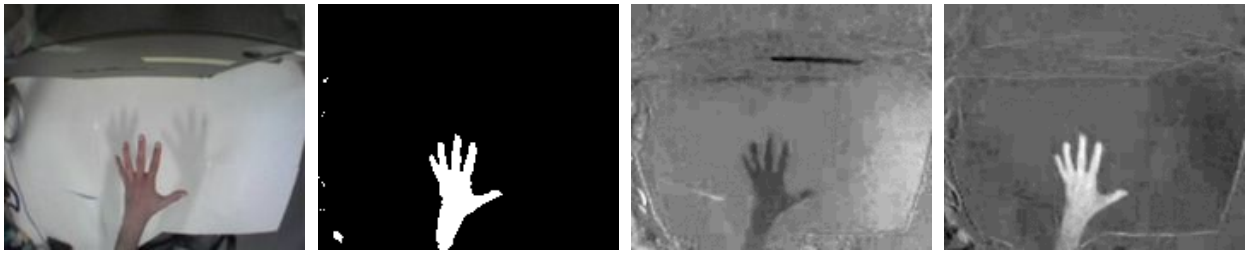
$$\forall(x,y), \quad S(x,y) = 1 \text{ si } \begin{cases} 77 \leq C_b \leq 127 \\ 133 \leq C_r \leq 173 \end{cases} \tag{4.11}$$

La figure 4.3 montre que le résultat est satisfaisant même si certains pixels de peau ne sont pas détectés et qu'il y a des fausses détections.

4.2.3.2 Histogramme C_bC_r

Nous avons vu en introduction de cette section qu'une des méthodes les plus performantes consiste à modéliser la couleur de peau par un histogramme dans l'espace C_bC_r, et à utiliser une règle de décision bayésienne, afin de classer les pixels peau et les pixels non-peau.

1. *Expectation-Maximisation*



(a) *image 370* (b) *résultat* (c) C_b (d) C_r

FIGURE 4.3 – *Exemple de segmentation de la main avec les seuils C_bC_r.*

Les densités de probabilité sont estimées avec l'histogramme de la couleur de peau h_{peau}, et l'histogramme total h_{total}, tous deux calculés dans l'espace C_bC_r lors d'une phase d'apprentissage, avec M images sélectionnées où les pixels ont été étiquetés comme appartenant aux classes peau ou non-peau.

Les histogrammes permettent d'utiliser la règle de BAYES pour calculer la probabilité qu'un pixel $\mathbf{x}$ soit un pixel de classe peau :

$$p\left(peau \mid \mathbf{x}\right) = \frac{p\left(peau\right)}{p\left(\mathbf{x}\right)} p\left(\mathbf{x} \mid peau\right) \tag{4.12}$$

Avec M images de $h \times l$ pixels, on a $N = M \times h \times l$ pixels parmi lesquels N_{peau} sont des pixels de classe peau. On peut donc écrire les probabilités suivantes à partir des histogrammes :

$$p\left(peau\right) = \frac{N_{peau}}{N} \tag{4.13}$$

$$p\left(\mathbf{x}\right) = \frac{h_{total}(\mathbf{x})}{N} \tag{4.14}$$

$$p\left(\mathbf{x} \mid peau\right) = \frac{h_{peau}(\mathbf{x})}{N_{peau}} \tag{4.15}$$

Ce qui donne après simplification :

$$p\left(peau \mid \mathbf{x}\right) = \frac{h_{peau}(\mathbf{x})}{h_{total}(\mathbf{x})} \tag{4.16}$$

La carte de probabilité obtenue est ensuite seuillée à $0,5$ pour obtenir les régions correspondantes à la classe peau (figure 4.4).

PAS D'ÉCHANTILLONNAGE

Un paramètre important de cette approche est le nombre de valeurs utilisées pour construire l'histogramme. Une taille plus importante permet une représentation plus fine de l'histogramme, mais nécessite une quantité de mémoire plus importante.

PHUNG *et al.* [107] montrent que les performances (courbes ROC [2]) sont meilleures en utilisant 256×256 valeurs, cependant la différence avec 128×128 et 64×64 valeurs est très faible. Avec 32×32 valeurs, les performances sont un peu moins bonnes, mais la différence est encore faible. Comme attendu, ils confirment que pour des pas d'échantillonnage plus faibles, il est nécessaire d'avoir un plus grand nombre d'images pour l'apprentissage, afin que l'histogramme ne soit pas trop bruité.

2. *Receiver Operating Characteristic*

(a) *image 370* (b) *carte de probabilité* (c) *résultat*

FIGURE 4.4 – *Segmentation de la main par histogramme C_bC_r (64×64 valeurs).*

APPRENTISSAGE DES HISTOGRAMMES

Un inconvénient des méthodes basées sur la modélisation de la couleur de la peau est qu'elles nécessitent une phase d'apprentissage, à partir d'images sélectionnées, dans lesquelles la main est segmentée, ce qui est parfois fait manuellement. Afin d'automatiser cette étape d'apprentissage, nous la réalisons en utilisant le résultat de la segmentation avec les seuils sur les composantes C_b et C_r (paragraphe 4.2.3.1), sur les premières images du flux vidéo. Le résultat de segmentation des seuils est utilisé pour calculer les histogrammes, avec N images.

MISE À JOUR DES HISTOGRAMMES

Pour adapter le modèle de la couleur de la peau aux effets des changements d'illumination, il est possible d'actualiser les histogrammes. Nous proposons de les mettre à jour périodiquement, pour prendre en compte l'évolution de la luminosité. Il existe des méthodes plus complexes, comme celle proposée par SIGAL *et al.* [119], qui modélisent la dynamique de la distribution de peau avec un modèle de MARKOV.

4.2.4 *Post-traitements*

Les méthodes présentées dans ce chapitre agissent exclusivement au niveau pixel. Par conséquent, il n'est pas garanti que les pixels détectés comme appartenant à un objet soient connectés entre eux. De plus, il y a généralement des fausses détections, qui se manifestent par des pixels blancs isolés, ainsi que des pixels de l'objet non détectés, qui se manifestent par des « trous » dans le masque. Il est donc nécessaire de filtrer l'image binaire pour ne garder que la région correspondant à la main.

Nous utilisons d'abord un *filtre médian* 5×5 pour supprimer les pixels isolés ; puis une *ouverture morphologique* pour connecter des régions proches et boucher les trous à l'intérieur de celles-ci. Après ces opérations, il peut rester des trous dans le masque (figure 4.5c) ; il peut aussi rester des objets non désirés, à cause d'une mauvaise segmentation. Un étiquetage en *composantes connexes* est réalisé pour regrouper les pixels adjacents du premier plan, en leur attribuant une même étiquette. Il est ensuite possible de supprimer les petites composantes connexes, pour ne garder que celle de la main (qui dans notre cas est la composante de surface la plus importante). Pour résoudre le problème des trous à

(a) *image* (b) *médian 5×5* (c) *ouverture*

FIGURE 4.5 – *Exemple de filtrages, sur une image binaire de la main. L'élément structurant utilisé pour les opérations morphologiques est un carré 3×3.*

l'intérieur de la main, un étiquetage du fond est réalisé, afin de supprimer les petites composantes correspondant aux trous.

4.2.5 *Suivi par boîte englobante*

Pour réduire les temps de calcul, il est possible de restreindre les traitements à une région d'intérêt contenant la main. Pour réaliser ceci, nous utilisons un algorithme de suivi par boîte englobante.

La boîte englobante d'un objet est simplement le plus petit rectangle contenant cet objet. Lorsque plusieurs objets sont présents dans la scène, le suivi consiste à mettre en correspondance les objets d'une image à la suivante, en utilisant le taux de recouvrement des boîtes englobantes. Cette méthode est par contre très sensible au bruit et aux objets multiples. Si deux objets se superposent dans le plan image, ils forment une seule région et il devient impossible de les distinguer.

Nous utilisons la boîte englobante pour prédire la position de la main dans l'image suivante : une *fenêtre de recherche*[3] est déduite de la boîte englobante obtenue à l'image précédente, en la multipliant par un coefficient de dilatation. Pour prendre en compte les variations de taille de la main, le coefficient de dilatation est calculé à partir du rapport entre la surface de la boîte englobante dans l'image courante et la surface dans l'image précédente.

Cette fenêtre de recherche permet de restreindre les traitements (segmentation, extraction de caractéristiques) à une zone de l'image, et donc de diminuer le temps de calcul. Il faut toutefois faire attention à ne pas perdre l'objet, dans le cas d'un mouvement trop rapide, ou à ne pas en « couper » une partie. Pour éviter ce genre de problème, des critères sont utilisés sur le rapport entre la surface de la main et la surface de la fenêtre de recherche. Si ce rapport est trop élevé, il faut augmenter la surface de la fenêtre ; s'il est trop faible, il est possible que la main soit perdue ou qu'il en manque une partie, et il est alors préférable de retraiter l'image entière.

4.2.6 *Algorithme utilisé*

L'algorithme de la figure 4.6 résume les différentes étapes pour la segmentation de la couleur de peau. L'apprentissage est réalisé automatiquement avec les N premières images du flux vidéo dans lesquelles la main apparaît : les

3. *Region Of Interest* (ROI)

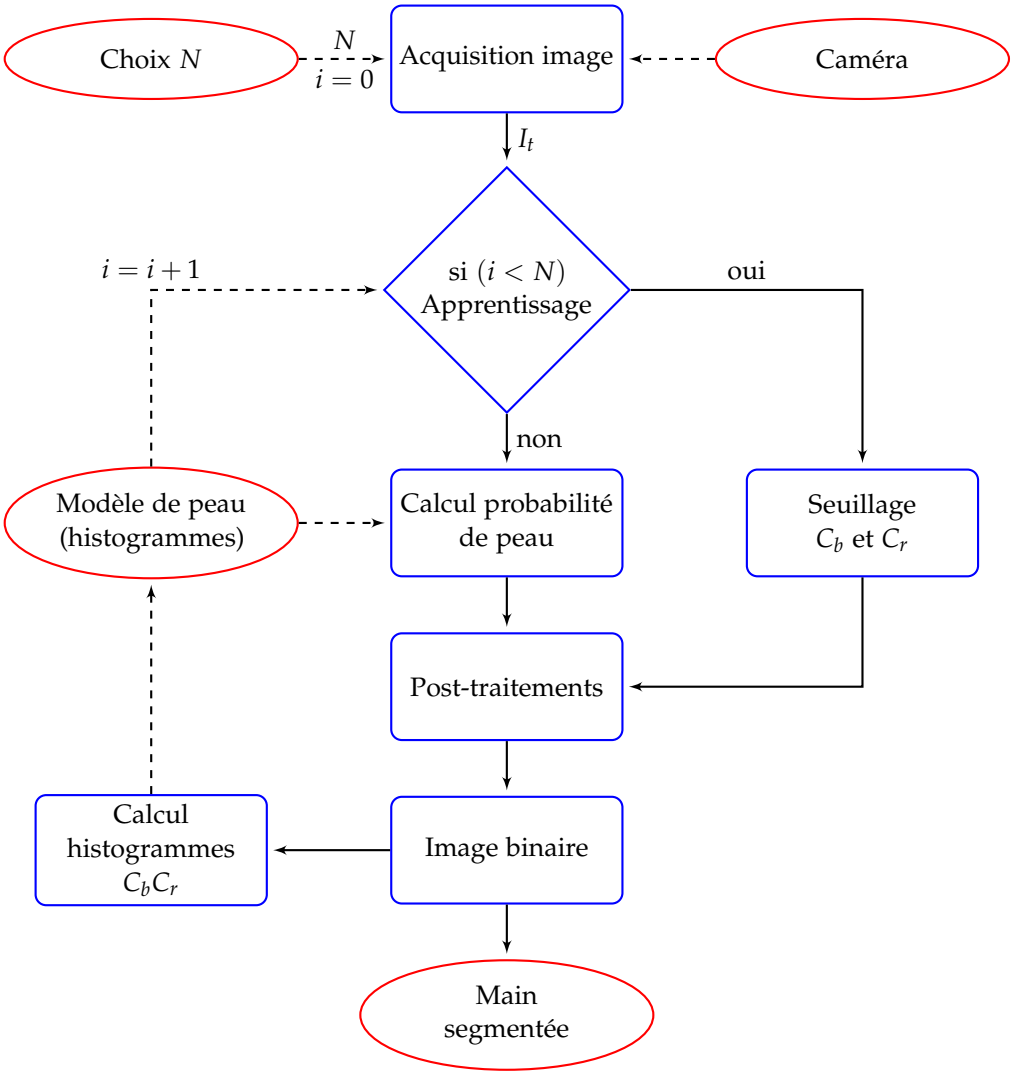


FIGURE 4.6 – *Algorithme de segmentation de la couleur de peau, N est le nombre d'images utilisées pour l'apprentissage.*

seuils sur les composantes C_b et C_r sont utilisés pour détecter la main et calculer les histogrammes modélisant la couleur de la peau. Si la surface détectée est inférieure à une valeur donnée, par exemple 50 pixels, on considère que la main n'est pas détectée et l'algorithme tourne en boucle. Une fois les histogrammes initialisés, ils sont ensuite utilisés pour calculer la probabilité que chaque pixel soit un pixel de peau. Le modèle de couleur de la peau est mis à jour périodiquement. Tous les traitements sont réalisés uniquement dans la fenêtre de recherche calculée à partir de la boîte englobante.

Le nombre d'images N pour l'apprentissage et la période de mise à jour est de l'ordre de 50 à 100. Pour l'apprentissage, le nombre d'images doit être suffisamment élevé pour que l'histogramme calculé soit représentatif de la distribution de peau. Ce nombre d'images dépend donc du pas d'échantillonnage de l'histogramme. Pour la mise à jour, le nombre d'images doit être choisi pour que l'algorithme s'adapte correctement aux variations de luminosité. Si ces variations sont faibles, la fréquence de mise à jour peut être faible, ce qui limite les calculs supplémentaires causés par une mise à jour de l'histogramme.

4.2.7 *Résultats et discussion*

Le tableau 4.1 montre les pourcentages de pixels non détectés et mal détectés par rapport à la surface de la main, obtenus sur une dizaine d'images pour lesquelles une segmentation de référence a été réalisée à la main. On constate que les meilleurs résultats sont obtenus avec l'histogramme C_bC_r à 64×64 valeurs avec mise à jour. La soustraction du fond présente un pourcentage de fausses détections très élevé, à cause des ombres, ce qui explique aussi que le pourcentage de pixels non détectés est très faible.

Les temps de calculs sont comparés avec les post-traitements. En effet, ceux-ci sont nécessaires pour l'apprentissage des histogrammes, afin d'éliminer les fausses détections. Ainsi, afin que la comparaison soit valable, tous les temps de calculs ont été mesurés avec les mêmes post-traitements. Ces temps de calculs sont relativement élevés, mais des optimisations sont possibles dans le cadre d'une application. La figure 4.7 permet de comparer visuellement le résultat brut (sans post-traitements) des différentes méthodes.

La segmentation par différence d'images est une approche très utilisée pour la détection et le suivi d'objets, mais celle-ci n'est pas adaptée à notre problème. En effet, dans notre configuration, la main provoque des ombres sur le plan de travail, qui sont segmentées avec la main. Pour utiliser la différence d'images, il nous faudrait utiliser une modélisation plus complexe du fond, avec un mélange de gaussiennes, et une méthode de détection des ombres. Mais cette approche est beaucoup plus coûteuse en temps de calcul.

La méthode des seuils sur les composantes C_b et C_r a l'avantage d'être très simple, ce qui donne un temps de calcul très faible. Toutefois, cette méthode est moins performante que pour celles reposant sur un apprentissage, le pourcentage de pixel non détectés étant plus élevé.

La méthode basée sur une modélisation par des histogrammes donne les meilleurs résultats, particulièrement avec 64×64 valeurs et la mise à jour périodique. De plus, la segmentation de peau comporte différents avantages sur la différence d'images :

- la caméra peut être en mouvement,
- l'utilisation de la chrominance améliore la robustesse aux variations de luminosité,
- les méthodes basées sur la couleur de peau sont insensibles aux ombres ainsi qu'aux variations du fond (déplacement d'un objet, de l'utilisateur ou d'une autre personne, etc.),
- cette méthode est presque instantanée puisqu'il suffit de regarder dans l'histogramme la valeur correspondant à la couleur de pixel.

Cependant, toutes ces méthodes possèdent un inconvénient commun : les objets dont la couleur est proche de celle de la peau sont aussi détectés. On parle alors de *camouflage* (cf. paragraphe 3.2.1).

Notre méthode permet d'automatiser l'apprentissage des histogrammes, en utilisant la détection de peau par des seuils, et de mettre à jour les histogrammes pour prendre en compte les variations de luminosité. Par conséquent, nous utiliserons cette méthode que la suite de nos travaux. L'algorithme utilisé est représenté au paragraphe 4.2.6.

Nous avons pu vérifier la robustesse de cet algorithme lors d'une démonstration du suivi de la main, à l'occasion de la « semaine des applications » au

MÉTHODE	NON DÉTECTÉS	FAUSSES DÉTECTIONS	TOTAL	TEMPS DE CALCUL
Soustraction du fond	0,7 %	21,5 %	22,2 %	31,4 ms
Seuil C_bC_r	11,2 %	7,4 %	18,6 %	22,1 ms
Histogramme 32×32	20,6 %	3,1 %	23,6 %	25,7 ms
Histogramme 64×64	4,1 %	21,9 %	26,0 %	25,9 ms
Histo 64 + MAJ	4,0 %	7,8 %	11,8 %	59,2 ms
Histo 64 + MAJ + ROI	3,9 %	7,7 %	11,7 %	25,1 ms

TABLEAU 4.1 – *Comparaison numérique des méthodes de segmentation avec les pourcentages de pixels non détectés, mal détectés et le total des deux, et les temps de calcul (en ms/image, avec un PC à 2 GHz). MAJ signifie que l'histogramme est mis à jour (toutes les 50 images), et ROI signifie que la segmentation est réalisée sur une région de l'image (« region of interest ») afin de réduire le temps de calcul.*

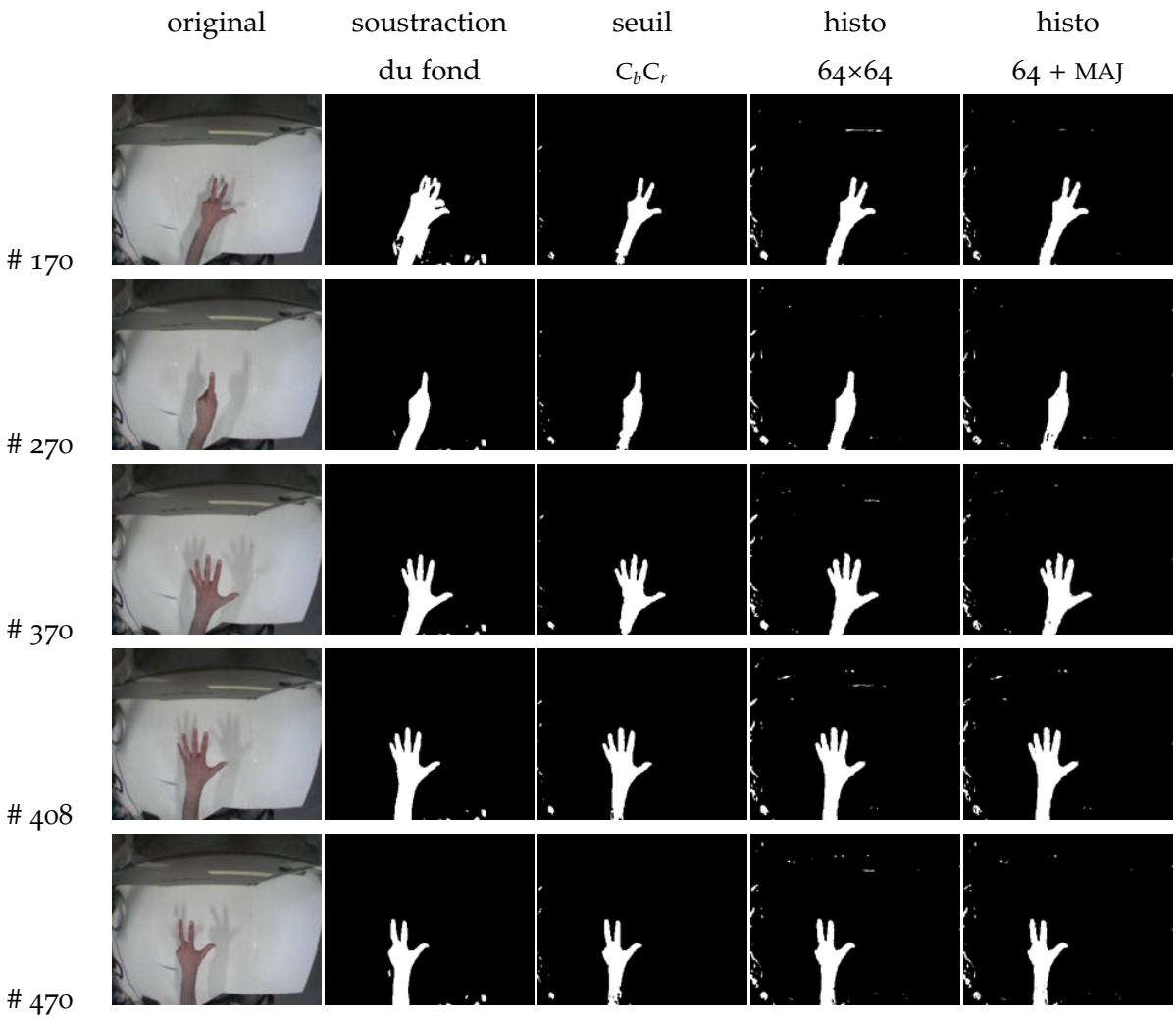


FIGURE 4.7 – *Comparaison en images des méthodes de segmentation : la soustraction du fond, le seuil sur les composantes C_bC_r, l'histogramme C_bC_r avec et sans mise à jour (avec 50 images pour l'apprentissage et une période de mise à jour de 50 images).*

sein de l'entreprise STMicroelectronics. Pendant une semaine, les différents produits développés par l'entreprise sont présentés aux employés. Cette démonstration se déroule dans une salle, avec par conséquent un environnement contrôlé. Ainsi, plusieurs dizaines de personnes ont pu tester le système et donc la segmentation de la couleur de la peau (ainsi que la reconnaissance de postures et le suivi de la main présentés dans les chapitres suivants). Nous avons pu constater que notre algorithme de segmentation se comporte bien dans un environnement intérieur, sur de nombreux types de peaux différents. Toutefois, pour des peaux très claires ou très foncées, les valeurs de chrominance sont moins significatives, et la segmentation est alors de moins bonne qualité.

4.3 EXTRACTION DE CARACTÉRISTIQUES MORPHOLOGIQUES

Le *centre de la main* est un point important car il est souvent le premier à être localisé et est très utilisé pour le suivi de la main [58]. Il permet de représenter la position globale de la main, ce qui est parfois suffisant pour reconnaître certains gestes, tels que des trajectoires 3D ou des gestes de la langue des signes [91, 121]. Il est aussi très utile de connaître la position du centre pour détecter les bouts des doigts, avec par exemple la distance au centre [140].

Un des problèmes pour la détection du centre est de savoir si l'utilisateur a le bras nu ou non. En effet, si celui-ci est nu, la détection de la couleur de peau fournit une région contenant la main et l'avant-bras. Cela pose aussi des problèmes pour la détection des doigts et le calcul des caractéristiques de forme. Ce qui nous amène à la nécessité de *détecter le poignet* pour séparer la main de l'avant-bras, afin d'éviter d'imposer des contraintes aux utilisateurs telles que le port d'un vêtement à manches longues.

Les *bouts des doigts* sont des points très importants pour le suivi et la reconnaissance de gestes, aussi bien avec des modèles 3D qu'avec des modèles d'apparence. Pour les gestes de pointage, le doigt est un moyen simple et intuitif pour remplacer la souris, pour pointer sur un écran ou sur des surfaces interactives. Pour obtenir la position du doigt avec une seule caméra, Wu *et al.* [136] utilisent un modèle géométrique du bras et calculent la direction de pointage avec la droite de vue entre l'œil et la main. D'autres méthodes utilisent la vision stéréoscopique pour calculer la position 3D du doigt, et éventuellement la direction de pointage [114].

Les bouts des doigts peuvent aussi être utilisés pour la reconnaissance de gestes simples en comptant le nombre de doigts [90], ou pour recaler un modèle 3D en minimisant la distance entre les doigts détectés et les projections du modèle dans l'image [86, 111]. Toutefois, nombre de modèles n'utilisent que les emplacements 2D des bouts des doigts et de la paume, d'où une forte dépendance au point de vue de la caméra.

Bien que les bouts des doigts soient fréquemment utilisés, il n'est pas trivial de les localiser précisément sans l'utilisation de gants ou de marqueurs colorés (Davis et Shah [35]). Il existe de nombreuses méthodes pour détecter les doigts, telles que celles basées sur le contour, que nous présentons au paragraphe 4.3.3, ou celles utilisant des modèles de bout de doigts. Par exemple, Lin *et al.* [88] localisent les bouts des doigts avec des modèles géométriques, constitués par des motifs de pixels représentant la forme en « U » des pixels du bout des doigts.

Une autre méthode basée sur un modèle de doigt est la mesure de corrélation (cf. section 3.3).

4.3.1 *Centre de la main*

Le centre de la main est primordial pour le suivi, ainsi que pour la détection des doigts. En effet, les bouts des doigts sont les points du contour qui se trouvent aux extrémités de la région de la main. Ainsi, il est possible de détecter ces points en utilisant des connaissances a priori sur la morphologie de la main, par exemple en calculant la distance par rapport au centre de la main.

Nous proposons deux méthodes pour calculer le centre. La première est classique, et consiste à calculer le centre de gravité à partir des moments géométriques. La deuxième méthode est basée sur le calcul d'une carte de distance.

4.3.1.1 Avec les moments géométriques

Les moments géométriques d'une image binaire I sont une des méthodes les plus simples pour décrire un objet. En se basant sur la formulation classique des moments bi-dimensionnels, un moment d'ordre $p+q$ s'écrit :

$$m_{pq} = \sum_{(x,y)\in I} x^p\, y^q\, I(x,y), \quad (p,q) \in \mathbb{N}^2 \tag{4.17}$$

Le moment d'ordre 0, m_{00}, donne la surface de la main. Le *centre de gravité* (x_G, y_G) de la région de la main peut être calculé à partir des moments d'ordre 1 de cette région :

$$(x_G, y_G) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{4.18}$$

L'inconvénient majeur de cette méthode est que, si la région détectée contient aussi l'avant-bras de l'utilisateur, alors le centre de gravité de la région ne correspond pas véritablement au centre de la main. Il se retrouve déplacé vers l'avant-bras, avec des variations suivant que le poing est fermé ou que les doigts sont ouverts. Pour utiliser cette méthode, une étape supplémentaire de séparation entre l'avant-bras et la main est donc nécessaire (paragraphe 4.3.2).

Les moments géométriques permettent aussi de calculer des invariants, ce que nous verrons au paragraphe 5.2.1, et de déterminer l'orientation de la région.

ORIENTATION

Les moments du second ordre permettent de calculer les axes d'inertie de la région à partir des moments géométriques d'ordre 2, grâce aux trois variables intermédiaires suivantes :

$$\begin{aligned} a &= \frac{m_{20}}{m_{00}} - x_G^2 \\ b &= 2\left(\frac{m_{11}}{m_{00}} - x_G y_G\right) \\ c &= \frac{m_{02}}{m_{00}} - y_G^2 \end{aligned}$$

L'orientation θ et les dimensions l_1 et l_2 d'un rectangle ayant les mêmes moments que notre forme sont alors données par :

$$\theta = \frac{1}{2} \arctan\left(\frac{b}{a-c}\right) \tag{4.19}$$

$$l_1 = \frac{a+c+\sqrt{b^2+(a-c)^2}}{2} \tag{4.20}$$

$$l_2 = \frac{a+c-\sqrt{b^2+(a-c)^2}}{2} \tag{4.21}$$

Le résultat est illustré par la figure 4.8.

4.3.1.2 Avec une carte de distance

Cette méthode se base sur l'idée que le centre de la main est le point situé à la plus grande distance des bords de la main. Nous proposons donc de localiser le centre avec une *carte de distance*, obtenue par le calcul d'une *transformée en distance* sur l'image binaire de la main.

La transformée en distance d'une image binaire associe à chaque pixel de l'objet la distance au pixel du contour le plus proche (figure 4.9). Une carte de distance est associée à une métrique (distances euclidienne, de Manhattan). Cette métrique modifie le poids donné aux pixels verticaux, horizontaux, et diagonaux. La carte de distance, que l'on peut voir sur la figure 4.10, peut aussi être utilisée pour la squelettisation. Le centre de la main est déterminé comme étant le maximum de la carte de distance.

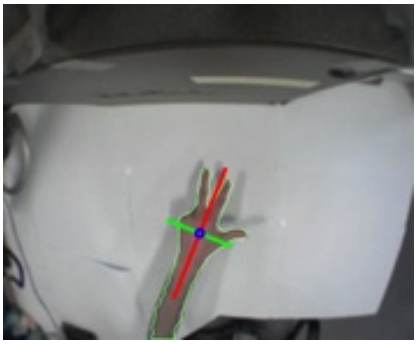


FIGURE 4.8 – *Détection du centre avec la carte de distance (cercle bleu), et de l'orientation et des axes d'inertie de la région de la main avec les moments géométriques.*

4.3.2 *Détection du poignet*

Si l'utilisateur a le bras nu, l'image binaire obtenue par segmentation contiendra une région avec la main et l'avant-bras. Or, pour certaines applications comme la reconnaissance de gestes, avoir l'avant-bras dans la région détectée peut fausser les résultats. En effet, dans ce cas le contour obtenu ne représente pas exactement la forme de la main. Il faut donc trouver une méthode pour séparer la main de l'avant-bras, au niveau du poignet.

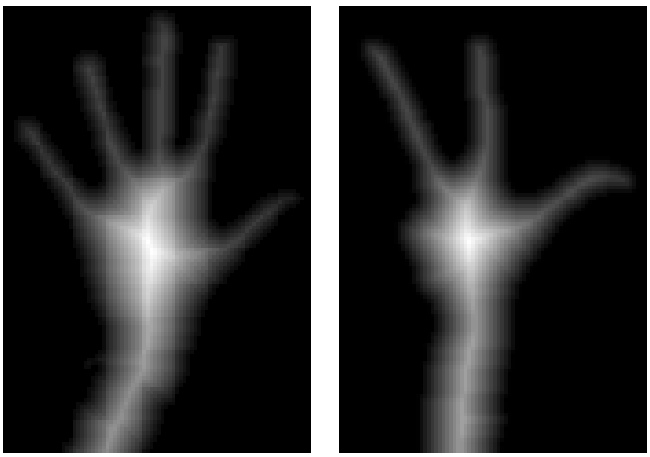
0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

(a) *image binaire*

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

(b) *carte de distance*

FIGURE 4.9 – *Calcul de la carte de distance d'une forme simple.*



(a) (b)

FIGURE 4.10 – *Exemple de cartes de distance, calculées à partir d'images binaires de la main.*

4.3.2.1 Méthodes existantes

Pour simplifier le problème, il est bien sûr possible de demander à l'utilisateur de porter un vêtement avec des manches ou de lui faire porter une bande colorée au niveau du poignet (LOCKTON ET FITZGIBBON [89]). Ainsi, pour l'acquisition de notre base de données de gestes (section 2.5), les utilisateurs portaient des manches afin de s'affranchir du problème de la détection du poignet. Toutefois, ces solutions ne sont pas satisfaisantes pour l'utilisateur. Il est préférable de trouver une méthode automatique de traitement d'images pour détecter le poignet.

WAH NG ET RANGANATH [132] proposent deux méthodes. La première est basée sur le calcul de la largeur de l'avant-bras, avec l'idée que cette largeur est relativement constante et augmente subitement lorsqu'on arrive à la main. La deuxième méthode se base sur la courbure du contour, qui augmente entre le poignet et le pouce, ce qui suppose que le pouce soit présent. Toutefois, ils ne fournissent aucun détail concernant la mise en oeuvre de ces méthodes.

De la même façon, LICSAR ET SZIRANYI [87] calculent la largeur de l'avant-bras, dans un axe perpendiculaire à l'orientation de l'avant-bras, fournie par les moments.

Une autre solution, proposée par CHEN *et al.* [20], consiste à comparer la position du centre de la main (obtenue lors de l'étape de segmentation) et celle de la boîte englobante de la région. Des critères morphologiques sont alors utilisés pour rogner la boîte englobante, jusqu'à ce qu'elle ne contienne que la main.

4.3.2.2 Méthode proposée

Nous avons testé différentes solutions, inspirées des méthodes existantes, mais ces méthodes ne sont pas toujours satisfaisantes, notamment dans les configurations où le poing est fermé. En effet, la variation de largeur entre l'avant-bras et la main est faible, et les méthodes basées sur la largeur du bras sont prises en défaut.

La solution que nous proposons est une combinaison de deux méthodes. La première consiste à réduire la taille de la boîte englobante en utilisant la position du centre de la main, détectée avec la carte de distance. La boîte englobante est centrée verticalement sur le centre de la main, et sa hauteur est fixée à deux fois la distance entre le centre et l'extrémité de la main (figure 4.11a). La deuxième consiste à mesurer la largeur de l'avant-bras, en parcourant le contour, et à détecter une augmentation de cette largeur (figures 4.11b et 4.11c). La position du poignet est détectée avec la dérivée de la courbe de distance. Dans le cas où le pouce est présent, cette augmentation est facilement détectable, et correspond bien au poignet. Mais si le pouce n'est pas présent, la variation de largeur est plus faible et pas toujours détectable ; d'où l'intérêt de la combinaison avec la première méthode pour ce cas de figure.

La figure 4.12 montre que notre méthode permet bien de détecter le poignet, quelle que soit la configuration de la main, notamment lorsqu'elle est fermée (figure 4.12b).

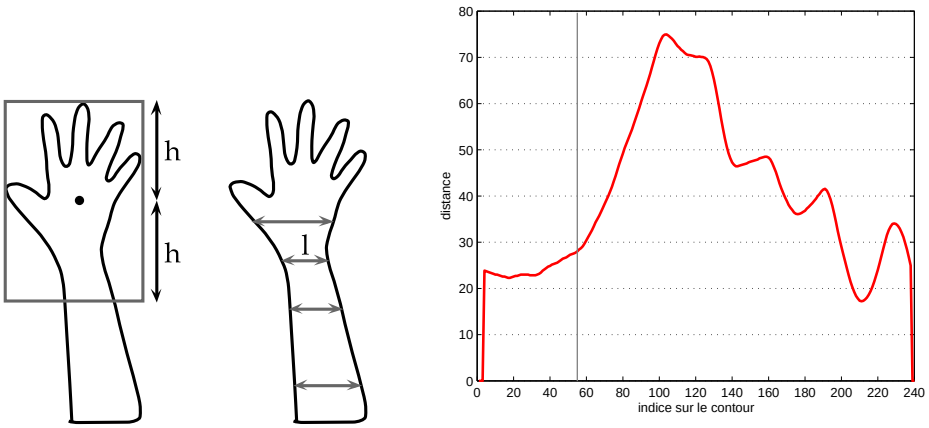
4.3.3 *Détection des doigts*

Les bouts des doigts sont des points caractéristiques très utilisés pour le suivi de la main et la reconnaissance de gestes. Ils ont l'avantage d'être facilement détectables, avec une complexité de calcul relativement limitée. Nous utiliserons ces positions dans les chapitres suivants, pour la construction d'un modèle 2D de la main, pour le suivi des doigts, et pour le recalage d'un modèle 3D.

Nous présentons deux approches, l'une basée sur la distance au centre de la main, et l'autre sur la courbure du contour. Ces méthodes permettent aussi de détecter les creux entre chaque doigt.

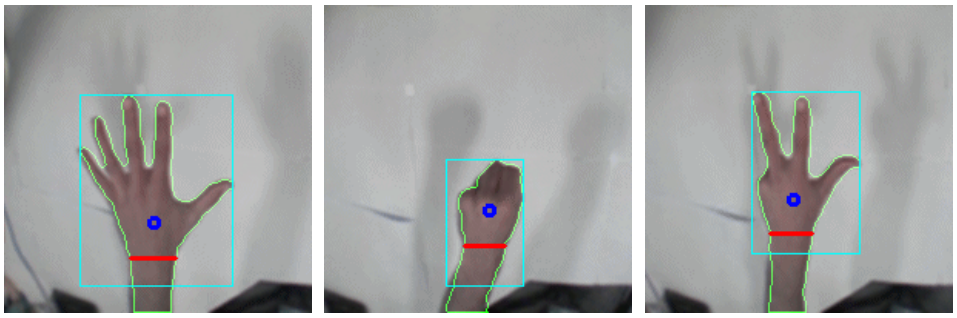
4.3.3.1 Avec la distance au centre de la main

Les bouts des doigts sont les points de contour qui se trouvent aux extrémités de la région de la main. Ainsi, on peut trouver ces points en calculant, pour



(a) (b) (c)

FIGURE 4.11 – *Détection du poignet pour séparer la main de l'avant-bras : (a) en réduisant la boîte englobante (avec le centre de la main détecté avec la carte de distance), (b) en mesurant la largeur de l'avant-bras, et (c) courbe de la largeur du poignet pour l'image 408 (cf. figure 4.15), le trait vertical correspond à la position détectée du poignet.*



(a) (b) (c)

FIGURE 4.12 – *Détection du poignet pour séparer la main de l'avant-bras : en vert le contour de la main, en cyan la boîte englobante réduite, et en rouge la coupure du poignet. Le rond bleu correspond au centre de la main, calculé avec la carte de distance.*

chaque point du contour P_i, la distance euclidienne $d(i)$ par rapport au centre de gravité G de la région correspondant à la main :

$$\forall i \in [1, N], \quad d(i) = D_{Eucl}(G, P_i) = \sqrt{(x_G - x_i)^2 + (y_G - y_i)^2} \tag{4.22}$$

Les maximums locaux de la courbe de distance obtenue correspondent aux bouts des doigts, et les minimums locaux correspondent aux creux. La figure 4.13 montre un exemple de courbe de distance pour une main ouverte (image 408 de la séquence utilisée pour les tests du paragraphe 4.3.4).

4.3.3.2 Avec la courbure du contour

La détection des bouts de doigts basée sur la courbure du contour est une méthode proposée par SEGEN ET KUMAR [114]. En effet, on peut voir les bouts des doigts comme les points du contour où la courbure est maximale. Une

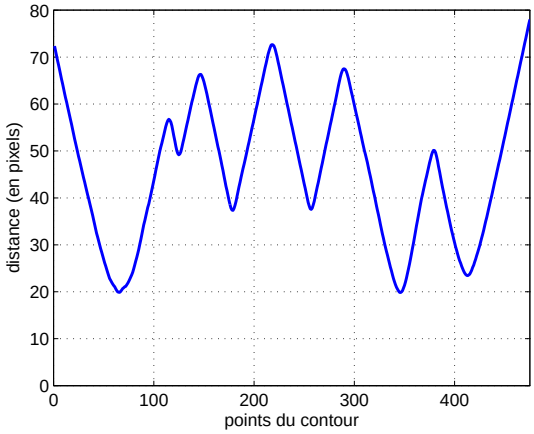
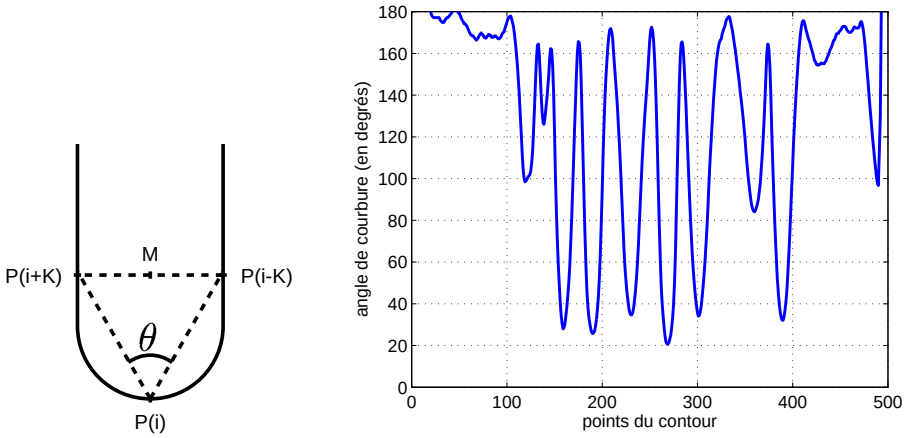


FIGURE 4.13 – *Courbe de la distance (en pixels) entre le centre de gravité de la main et les points du contour, pour une image de main avec cinq doigts.*



(a) (b)

FIGURE 4.14 – *Courbure du contour : (a) calcul de la k-courbure, et (b) tracé de l'angle de courbure θ avec $k = 10$ (en parcourant le contour dans le sens des aiguilles d'une montre).*

version simplifiée a été proposée par WU *et al.* [136], elle consiste à remplacer le calcul de la « *k-courbure* » par un produit scalaire.

Le contour est représenté par une suite de N points $\{P_i = (x_i, y_i)\}$, $1 \leq i \leq N$. La « *k-courbure* » est donnée par l'angle θ entre les vecteurs P_iP_{i-k} et P_iP_{i+k} :

$$\cos(\theta) = \frac{P_iP_{i-k}.P_iP_{i+k}}{\|P_iP_{i-k}\|.\|P_iP_{i+k}\|} \tag{4.23}$$

DISTINCTION ENTRE BOUT ET CREUX

Les minima locaux de la courbure (figure 4.14b) correspondent aux bouts des doigts, mais aussi aux creux entre les doigts. Il est donc nécessaire d'effectuer un test pour les différencier. Celui-ci consiste à regarder si M, le point milieu du segment entre P_{i-k} et P_{i+k} (figure 4.14a), est à l'intérieur de la région de la main. Si tel est le cas, il s'agit d'un bout de doigt, sinon c'est un creux.

PARAMÈTRE

Le paramètre k est choisi entre 10 et 20, en fonction de la surface de la région de la main. Si la surface est importante, k doit être élevé pour avoir une détection précise ; si elle est faible, k doit être petit pour être sûr de détecter tous les doigts. De manière générale, le paramètre k est choisi pour avoir un taux de détection élevé, quitte à obtenir des fausses détections. En effet, il est préférable d'être certain de bien détecter tous les doigts, les fausses détections pouvant être éliminées lors des étapes ultérieures.

4.3.4 *Résultats*

Ce paragraphe présente les résultats de détection du centre de la main, du poignet et des bouts des doigts, sur la séquence vidéo « `divers_L_1.avi` » présentée au paragraphe 2.5.1.

4.3.4.1 Centre de la main

La première colonne de la figure 4.15 montre le résultat de la détection du centre avec les moments géométriques et la carte de distance. On constate que la méthode basée sur les moments géométriques n'est pas fiable car elle dépend de la forme de la main, si elle est ouverte ou non, et surtout si l'avant-bras est aussi segmenté avec la main. Dans ce cas, le centre de gravité ne correspond pas au centre de la main. Contrairement à la première méthode, la carte de distance n'est pas influencée par la détection de l'avant-bras. Elle s'avère très robuste, la position du centre obtenue est fiable et bien stable dans le temps. C'est donc la carte de distance que nous utiliserons par la suite, notamment pour la détection des doigts.

4.3.4.2 Détection du poignet

La deuxième colonne de la figure 4.15 montre le résultat des deux méthodes de détection du poignet, utilisées séparément. On constate que dans certains cas la méthode de la boîte englobante fonctionne mieux (image 270), dans d'autres cas c'est celle basée sur la largeur du bras qui donne la meilleure position. La méthode de la boîte englobante est influencée par la présence des doigts, celle basée sur la largeur du bras est sensible aux défaut de segmentation du contour. En combinant ces deux méthodes, on obtient la meilleure des deux détections.

4.3.4.3 Détection des doigts

Le tableau 4.2 montre les pourcentages de bout des doigts détectés, avec la distance au centre et la courbure ($k = 10$). On constate que la distance au centre donne de meilleurs résultats lorsqu'il y a cinq doigts. Dans ce cas, la courbure est mise en défaut lorsque l'annulaire est trop proche de l'auriculaire. Les taux de fausses détections sont très faibles pour les deux méthodes (inférieur à 1%).

La dernière colonne de la figure 4.15 montre le résultat de détection des bouts des doigts, avec la courbure et la distance au centre de la main. Les deux méthodes donnent des résultats très similaires, même si l'on peut constater que la courbure est parfois moins précise (image 408) car elle est plus sensible aux variations du contour résultant d'une mauvaise segmentation. Pour atténuer

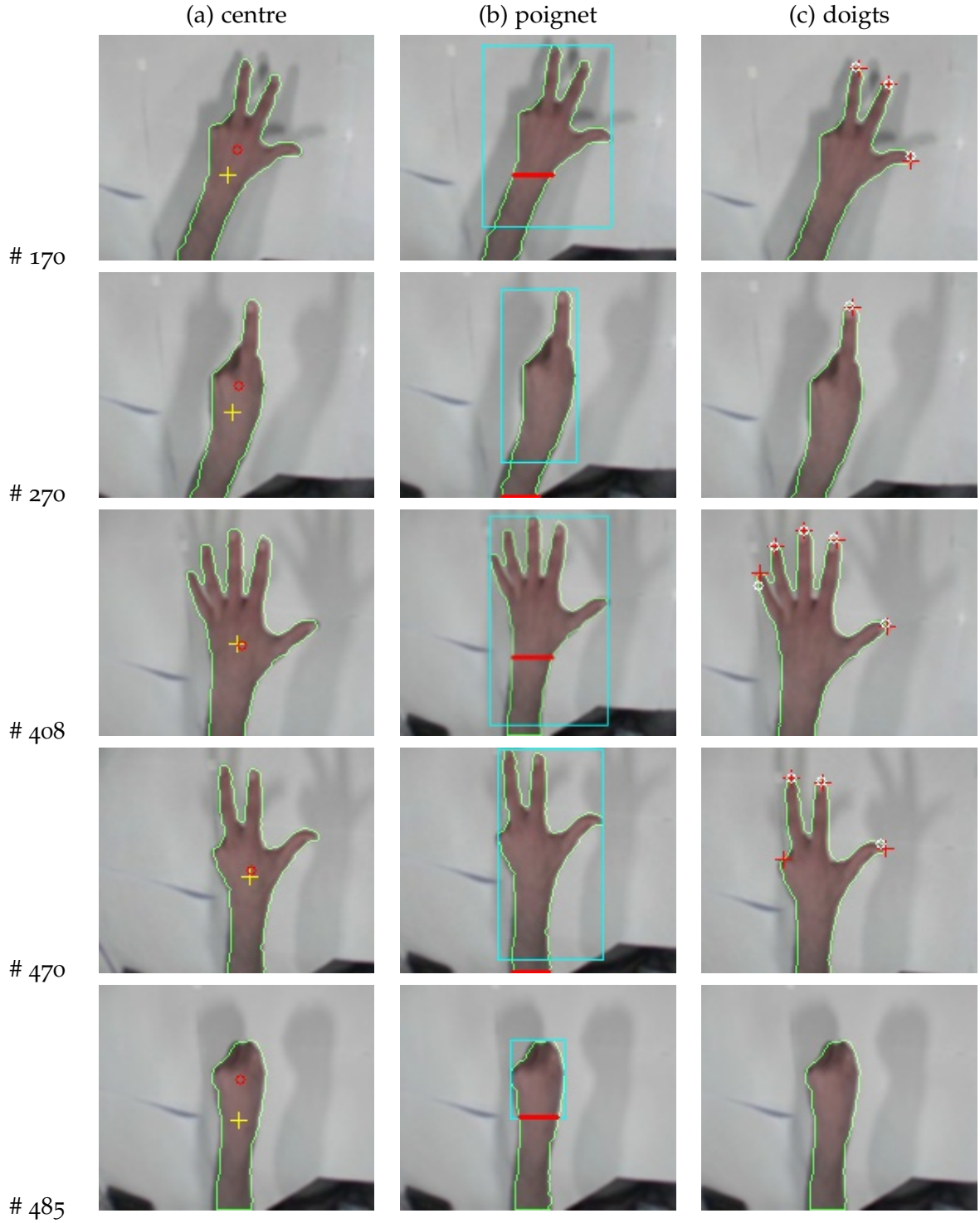


FIGURE 4.15 – *Comparaison des méthodes de détection sur la séquence vidéo « `divers_L_1.avi` » (avec des images découpées et agrandies pour bien visualiser les détections) : (a) centre de la main avec les moments géométriques (croix jaune) et la carte de distance (cercle rouge), (b) poignet avec la boîte englobante (rectangle cyan) et la largeur du bras (trait rouge) séparément, et (c) bouts des doigts avec la courbure (cercle blanc) et la distance au centre (croix rouge).*

SÉQUENCES	DISTANCE	COURBURE
1 et 2 doigts	99,2 %	98,9 %
3 doigts	99,7 %	99,7 %
5 doigts	96,1 %	90,1 %
Total	97,1 %	93,1 %

TABLEAU 4.2 – *Comparaison des méthodes de détection de doigts avec la distance et la courbure (pourcentage de doigts détectés).*

l'effet de ces déformations, les courbes de courbure et de distance sont lissées avec un filtre moyenneur. De plus, la courbure repose sur le paramètre k, qui nécessite d'être adapté en fonction de la surface de la région de la main, ce qui représente un inconvénient pour une application. Par ailleurs, la courbure nécessite une étape supplémentaire pour différencier les bouts et les creux.

C'est donc la détection des doigts avec la distance au centre qui sera utilisée par la suite. Enfin, pour éliminer les fausses détections lorsque le point est fermé, nous utilisons le critère suivant : si le rapport entre la distance au bout de doigt et la distance au creux correspondant est inférieur à un seuil (typiquement 2), alors le doigt n'est pas détecté.

4.4 RÉSUMÉ

Dans ce chapitre, nous avons présenté différentes méthodes de segmentation de la main et d'extraction de caractéristiques morphologiques. Nous avons vu qu'il n'est pas évident de trouver une méthode de segmentation qui soit performante, robuste aux variations de luminosité ou à un fond complexe, et qui respecte la contrainte du temps réel. Or, la segmentation est une étape primordiale pour la suite des traitements.

La soustraction du fond nécessite une caméra fixe, et est très sensible aux ombres ainsi qu'aux variations de luminosité. Pour résoudre ce problème, au moins partiellement, il est possible d'utiliser une modélisation plus complexe du fond, avec des mélanges de gaussiennes. Mais ces méthodes nécessitent un temps de calcul trop élevé pour notre application, sans que le problème des ombres ne soit résolu de façon satisfaisante.

Les méthodes basées sur la couleur de peau sont beaucoup plus adaptées à notre problématique. Nous avons présenté deux méthodes : la première est basée sur un seuillage des composantes de chrominance, C_b et C_r. Elle permet une détection très rapide de la main, pour une qualité de segmentation satisfaisante dans la majeure partie des cas. La deuxième méthode est basée sur la modélisation de la couleur de peau par des histogrammes C_bC_r.

Nous avons proposé une méthode pour automatiser la phase d'apprentissage des histogrammes, en utilisant le résultat de la détection avec les seuils, et nous avons vu qu'il est possible de mettre à jour le modèle périodiquement, pour l'adapter aux variations de luminosité. Une démonstration réalisée au sein de l'entreprise STMICROELECTRONICS a permis de tester l'algorithme de segmentation avec de nombreuses personnes. Cette méthode fonctionne pour

les différents types de couleurs de la peau, mais risque cependant d'être mise en défaut sur des peaux très claires ou très sombres, ou si le fond a une couleur proche de la couleur de la peau. Dans le chapitre 6, nous utiliserons cette méthode de segmentation pour chaque caméra.

Nous avons ensuite présenté différentes méthodes d'extraction de caractéristiques de position à partir de l'image binaire ou du contour de la main. Le centre, le poignet et les bouts des doigts sont détectés en utilisant des connaissances a priori sur la morphologie de la main. Nous avons proposé une méthode robuste pour détecter le centre, basée sur le calcul d'une carte de distance. Celle-ci a l'avantage de ne pas être perturbée par la présence de l'avant-bras. En effet, nous avons vu que dans le cas où l'avant-bras est nu, il est segmenté avec la main, ce qui peut poser problème pour extraire des caractéristiques de forme de la main. Nous avons donc proposé une méthode pour détecter le poignet, quelle que soit la configuration de la main, et ainsi séparer la main de l'avant-bras. Pour finir, nous avons étudié deux méthodes pour détecter les bouts des doigts. Celle basée sur la distance au centre de la main s'avère être plus performante que celle basée sur la courbure du contour. Nous utilisons ces points caractéristiques dans le chapitre 6, pour le suivi de la main. Le chapitre suivant présente la comparaison de descripteurs de formes pour la reconnaissance de postures.

5

RECONNAISSANCE DE POSTURES DE LA MAIN

Dans ce chapitre, nous étudions la reconnaissance de postures de la main avec des descripteurs de forme, permettant de calculer un vecteur de caractéristiques représentant la forme de la main. Les gestes à reconnaître sont classifiés par rapport aux modèles calculés lors d'une étape d'apprentissage. Nous utilisons d'abord des bases de données d'images, afin de comparer les descripteurs et obtenir des résultats reproductibles, puis des flux vidéos.

Les différents descripteurs de forme que nous comparons sont les moments de Hu, les moments de Zernike et les descripteurs de Fourier (FD). Pour les descripteurs de Fourier, nous évaluons deux familles d'invariants : une famille « classique » (FD1), utilisant le module des coefficients de Fourier ; et une famille complète et stable (FD2), proposée par Ghorbel [48], qui doit théoriquement donner de meilleurs résultats.

Les gestes à reconnaître sont définis par un vocabulaire. Pour évaluer les performances des descripteurs face aux translations, rotations et changements d'échelle de la main, nous utilisons deux bases de données d'images, présentées dans la section 2.5 : celle de Triesch [126], ainsi que notre propre base. En effet, la base de Triesch étant limitée en nombre d'images et en variations de la forme des gestes, nous avons fait l'acquisition de notre propre base de données, ce qui nous permet d'obtenir des résultats dans les conditions de notre application. Nous avons ainsi défini un vocabulaire composé de 11 gestes, et collecté un grand nombre d'images en demandant à 18 personnes d'effectuer ces gestes.

Après avoir comparé les descripteurs de formes, nous évaluons plusieurs méthodes de classification : euclidienne, bayésienne, ainsi que les classifieurs k-NN et SVM. Enfin, nous proposons trois méthodes pour améliorer les résultats de reconnaissance de postures dans le cas du traitement d'un flux vidéo. La première consiste en un filtrage temporel, pour prendre en compte la stabilité temporelle lors de la réalisation d'un geste. La deuxième permet de rejeter les gestes « inconnus », qui ne correspondent à aucun geste du vocabulaire, ou les gestes « ambigus », notamment pour les transitions entre deux gestes. La troisième consiste à utiliser l'information supplémentaire fournie par une deuxième vue de la scène.

SOMMAIRE

5.1 INTRODUCTION

La reconnaissance de gestes est une tâche difficile, sujette à de nombreux travaux de recherche. Deux grandes approches peuvent être distinguées : celles basées sur un *modèle* 3D, et celles basées sur un modèle d'*apparence* de la main (cf. paragraphe 3.2.2.1). Les approches basées sur un modèle 3D permettent de retrouver la configuration exacte de la main, par recalage du modèle, sans nécessiter d'apprentissage. Mais ces méthodes sont très coûteuses en temps de calcul, souvent éloigné du temps réel.

Les approches par apparence sont plus dépendantes du point de vue de la caméra, elles sont aussi plus adaptées pour une reconnaissance en temps réel. Elles permettent de reconnaître un geste parmi un ensemble de gestes connus, appelé *vocabulaire*. Le modèle d'apparence de chaque geste est calculé à partir d'un ensemble d'images d'apprentissage. Les modèles d'apparence sont constitués de caractéristiques extraites des images, décrivant le contenu de l'image, ou d'imagettes extraites de l'image, par exemple pour la corrélation ou les approches par ACP [1].

Pour résoudre le problème de la dépendance au point de vue, les modèles d'apparence doivent posséder des propriétés d'invariances aux translations, rotations et changements d'échelle. Ces caractéristiques invariantes sont basées sur la région, pour les moments de HU [62] et les moments de ZERNIKE [113], ou sur le contour de la forme, pour les descripteurs de FOURIER ou les histogrammes d'orientation [43].

Habituellement, pour un système de reconnaissance de gestes, il faut définir un vocabulaire de postures de la main. La posture correspond à l'aspect statique du geste, à sa configuration à un instant donné, par opposition au côté dynamique du geste. Dans ce chapitre, nous parlons simplement de gestes.

Le vocabulaire de gestes est défini en choisissant un ensemble de gestes, et en prenant en compte certaines contraintes. Il existe des vocabulaires plus exhaustifs que d'autres, comme la langue des signes et les vocabulaires qui en sont inspirés (ONG ET RANGANATH [102]). Dans notre cas, nous désirons que les gestes soient simples et intuitifs pour l'utilisateur, facilement reproductibles, et suffisamment différents les uns des autres, afin de faciliter l'utilisation de ce vocabulaire par des personnes peu habituées à un tel langage.

5.1.1 *Descripteurs de* FOURIER

Les descripteurs de FOURIER (FD [2]) ont été popularisés par CRIMMINS [30] et PERSOON ET FU [106]. Ils sont largement utilisés pour la description et la classification de formes à contour fermé, car ils permettent une bonne représentation des formes et possèdent des propriétés d'invariance intéressantes. Les FD sont calculés à partir des coefficients de la transformée de FOURIER de la signature du contour.

Les descripteurs de FOURIER ont déjà été utilisés pour la reconnaissance de gestes [20, 87, 132]. Mais, dans ces travaux, les FD constituent une simple composante d'un système complet de reconnaissance. Ainsi, les performances des FD n'ont pas été analysées en détails, indépendamment des autres composantes

1. Analyse en Composantes Principales
2. *Fourier Descriptors*

du système. En règle générale, dans les travaux existants, la signature complexe est utilisée, ainsi que le module des coefficients de FOURIER (FD1). La deuxième famille de descripteurs (FD2) n'a pas été utilisée pour la reconnaissance de gestes. Ces points sont détaillés dans le paragraphe 5.2.3.

La classification est généralement réalisée avec une distance, ou avec les méthodes de type *plus proches voisins*. Le nombre d'images utilisé pour l'apprentissage est un facteur important pour la classification.

Les FD ont aussi été utilisés comme vecteurs caractéristiques pour la reconnaissance de gestes dynamiques, avec les HMM [3], ou avec les réseaux de neurones de type RBF [4]. Par exemple, HARDING ET ELLIS [52] utilisent les FD pour la reconnaissance de trajectoires 2D, obtenues avec des gestes de pointage.

CHEN *et al.* [20] utilisent les FD et une analyse du mouvement, pour reconnaître des gestes dynamiques avec les HMM. Leurs données de test sont constituées de 1 200 séquences, obtenues avec vingt gestes réalisés par vingt personnes. Ils obtiennent un taux de reconnaissance de 90,5%, et de 93,5% en combinant les FD avec des caractéristiques de mouvement.

WAH NG ET RANGANATH [132] utilisent les FD et un classifieur de type RBF pour reconnaître cinq postures. Ils proposent ensuite de reconnaître quatorze gestes dynamiques, dont certains sont réalisés avec les deux mains, avec des HMM ou des réseaux de neurones. Ils comparent les résultats avec les moments de ZERNIKE, utilisés jusqu'à l'ordre 22 : les performances sont comparables, mais le temps de calcul des moments de ZERNIKE est beaucoup plus élevé. Avec 692 images d'apprentissage et 329 images de test, les taux de classification sont d'environ 90%.

LICSAR ET SZIRANYI [87] proposent une méthode d'apprentissage interactif, avec des étapes supervisées et d'autres non-supervisées. Les paramètres des modèles de gestes sont actualisés avec les gestes bien classifiés. Si un geste est mal classifié, l'utilisateur peut envoyer un retour d'information au système, et recommencer l'apprentissage du geste.

Une étude récente sur la reconnaissance de postures du corps humain, réalisée par POPPE ET POEL [108], fournit une comparaison entre les FD avec la signature complexe, les moments de HU et les « *histogrammes du contexte de la forme* » [5]. Les auteurs réalisent des tests avec des formes déformées, pour évaluer la robustesse des différents descripteurs de formes. Ils montrent que les FD fournissent les meilleures performances.

5.1.2 *Autres méthodes*

De nombreuses autres méthodes ont été étudiées pour la reconnaissance de gestes. TRIESCH ET VON DER MALSBURG [126], dont nous utilisons la base de données, utilisent un graphe élastique pour représenter la forme de la main. Pour chaque geste, un graphe moyen est calculé à partir de plusieurs images, en utilisant des descripteurs locaux basés sur des filtres de GABOR, et un contrôle manuel. Cette méthode est invariante aux changements d'échelle et ne nécessite pas de segmentation.

3. *Hidden Markov Models*
4. *Radial-Basis Function*
5. *shape context histograms*

JUST *et al.* [71] utilisent un opérateur local non-paramétrique, la « *Transformée de Census modifiée* »[6], inspiré de travaux de recherche en détection de visage, et un classifieur Adaboost [45]. Ils effectuent des tests de classification et de reconnaissance avec la base de gestes de TRIESCH.

CAPLIER *et al.* [18] utilisent les moments de HU et un réseau de neurones « *perceptron multicouches* »[7], pour classer huit gestes réalisés par trois personnes. Ils étudient l'influence de la segmentation de la main, et l'apport que représente l'information tridimensionnelle avec un système d'acquisition spécifique.

KOLSCH ET TURK [77][78] proposent d'analyser les fréquences de l'image, pour améliorer le détecteur de VIOLA ET JONES, et obtenir ainsi un détecteur de la main dans des images en niveaux de gris, robuste à des fonds complexes et aux variations de luminosité. Ils appliquent leur méthode à la reconnaissance de six gestes.

MOGHADDAM ET PENTLAND [96] ont utilisé leur méthode d'estimation de l'espace des vecteurs propres sur des images de contour de la main. Ils proposent un modèle d'« *eigenhand* » permettant de localiser la main et de reconnaître des gestes.

5.2 CARACTÉRISTIQUES DE FORMES

Dans cette section, nous nous intéressons à l'extraction d'un vecteur de caractéristiques permettant de représenter la forme de la main. Étant donné que l'apparence de la main dans une image peut varier fortement en fonction du point de vue, pour une même configuration, nous cherchons des caractéristiques robustes à ce type de transformation. Normalement, il faudrait considérer la question des transformations affines, mais nous nous limitons au cas des transformations euclidiennes (translation, rotation, changement d'échelle) qui représentent la majeure partie des transformations auxquelles nous sommes confrontés.

Cette section présente les descripteurs de formes que nous utilisons pour la reconnaissance de gestes : les invariants de HU, les moments de ZERNIKE, et les descripteurs de FOURIER.

5.2.1 *Moments de* HU

Les moments de HU [62] constituent une famille d'invariants qui est utilisée de longue date pour la reconnaissance de formes. Les invariants sont calculés à partir des moments géométriques de l'image binaire de la main, présentés au paragraphe 4.3.1.1.

La connaissance du centre de gravité (x_G, y_G) de la région permet de calculer les *moments centrés*, μ_{pq} :

$$\mu_{pq} = \sum_{(x,y)\in I} (x - x_G)^p (y - y_G)^q \, I(x,y) \tag{5.1}$$

6. *Modified Census Transform* (MCT)
7. *multi-layer perceptron*

Les *moments centrés* sont invariants aux translations. Pour obtenir l'invariance aux changements d'échelle, on calcule les *moments normalisés*, η_{pq} :

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \text{avec} \quad \gamma = \frac{p+q}{2} + 1, \ \forall p+q \geq 2 \tag{5.2}$$

En utilisant les *moments normalisés*, jusqu'à l'ordre 3, on peut calculer les 7 moments invariants de HU :

$$I_1 = \eta_{20} + \eta_{02} \tag{5.3}$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{5.4}$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{5.5}$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{5.6}$$

$$\begin{aligned} I_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{5.7}$$

$$\begin{aligned} I_6 = {} & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})] \end{aligned} \tag{5.8}$$

$$\begin{aligned} I_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{5.9}$$

Les six premiers invariants caractérisent la forme avec l'invariance aux translation, rotation et changement d'échelle. Le septième invariant permet de distinguer des formes symétriques.

5.2.2 *Moments de* ZERNIKE

Les moments complexes de ZERNIKE (KHOTANZAD ET HONG [75]) sont construits autour d'une famille de polynômes complexes formant une base orthogonale, définie dans le cercle unité. Cette base orthogonale permet de réduire la redondance entre les moments. Des normalisations permettent d'obtenir une invariance de ces descripteurs aux transformations impliquant des rotations, translations et changements d'échelle.

Pour une image I, les moments de ZERNIKE s'écrivent de la façon suivante :

$$A_{mn} = \frac{m+1}{\pi} \sum_x \sum_y I(x,y) V_{mn}^*(x,y) \tag{5.10}$$

avec $x^2 + y^2 \leq 1$, $m = 0, 1, 2, ..., \infty$ est l'ordre des moments, n est un entier respectant les conditions suivantes :

$$\begin{cases} m - |n| & \text{est paire} \\ |n| \leq m \end{cases} \tag{5.11}$$

Les polynômes de ZERNIKE s'écrivent en coordonnées polaires :

$$V_{mn}(r,\theta) = R_{mn}(r) e^{jn\theta} \tag{5.12}$$

avec (r,θ) définis sur le disque unité. $R_{mn}(r)$ est le polynôme radial, il s'écrit de la façon suivante :

$$R_{mn}(r) = \sum_{s=0}^{\frac{m-|n|}{2}} (-1)^s \frac{(m-s)!}{s!(\frac{m+|n|}{2} - s)!\ (\frac{m-|n|}{2} - s)!} r^{m-2s} \tag{5.13}$$

Les modules des moments de ZERNIKE sont invariants aux rotations. Pour obtenir l'invariance en translation et changement d'échelle, les images sont normalisées en utilisant les moments d'ordre 0 et 1. Ainsi les moments $|A_{00}|$ et $|A_{11}|$ ne sont pas utilisés en reconnaissance. Selon KUMAR ET SINGH [83], il est suffisant pour la reconnaissance de prendre les moments d'ordre 2 à 15, ce qui représente 70 moments.

L'inconvénient majeur des moments de ZERNIKE est leur temps de calcul qui est très important. Différentes méthodes ont été proposées (HWANG ET KIM [65]) pour permettre un calcul plus rapide des moments. CHONG *et al.* [21] comparent différentes méthodes existantes et en proposent une qui permet de calculer les moments jusqu'à l'ordre 24 en 50 millisecondes au lieu de 1,10 secondes par la méthode directe, pour une image binaire de 50×50 pixels.

5.2.3 *Descripteurs de* FOURIER

Nous nous intéressons désormais aux descripteurs de formes calculés à partir d'un contour, pour obtenir des invariants aux similitudes. Une définition de la notion de forme a été proposée par GHORBEL ET DE BOUGRENET DE LA TOCNAYE [49] et MOKHTARIAN ET MACKWORTH [98] :

Soit $\mathcal{G}$ un groupe de transformations opérant sur un ensemble d'objets ou un ensemble de paramétrisations $\mathcal{X}$, et O_1, O_2 deux éléments de $\mathcal{X}$. On dit que O_1 et O_2 sont équivalents relativement à $\mathcal{G}$ s'il existe un élément $g \in \mathcal{G}$ tel que :

$$O_2 = g(O_1)$$

Il existe dans la littérature des méthodes pour obtenir des invariants aux similitudes. Parmi ces méthodes, nous nous intéressons aux descripteurs de formes, qui peuvent être calculés à partir d'une région (par exemple les moments), ou à partir d'un contour.

Le contour est calculé en parcourant l'image binaire ligne par ligne ou colonne par colonne, afin de localiser un point du contour : le *point de départ*. Un algorithme permet ensuite d'extraire un à un les points du contour de la forme. Puisque la forme peut changer de position ou orientation, le point de départ ne sera pas toujours le même, ce qui influe sur la représentation paramétrique du contour.

5.2.3.1 Paramétrage du contour

Nous nous intéressons essentiellement aux objets à contour fermé, représentés par le paramétrage suivant :

$$\gamma(t) = (x(t), y(t)), \quad t \in [0, T], \quad \gamma(0) = \gamma(T) \tag{5.14}$$

Considérons deux contours fermés de paramétrisations $\gamma_1(l)$ et $\gamma_2(l)$, ayant la même forme et des représentations différentes. De manière générale, on peut écrire :

$$\gamma_2(l) = ae^{j\theta}\gamma_1(l + l_0) + b \tag{5.15}$$

avec :

- a un facteur d'échelle,
- θ l'angle de rotation,
- b un vecteur de translation,
- l_0 la différence de point de départ entre les deux courbes.

5.2.3.2 Invariance

Le but de la description de formes est d'en extraire des propriétés caractéristiques invariantes, sous forme d'un vecteur de données.

Soit $\mathcal{G}$ un groupe de transformations opérant sur un ensemble $\mathcal{X}$. On appelle famille d'invariants sur $\mathcal{X}$ par rapport à $\mathcal{G}$ une famille de fonctions $\{I_n\}_{n\in\mathcal{J}}$ de $\mathcal{X}$ à valeurs scalaires, telle que pour deux objets O_1 et O_2 ayant la même forme, de paramétrisations γ_1 et γ_2, on a :

$$I_n(\gamma_2) = I_n(\gamma_1), \quad \forall n \in \mathcal{J}$$

Il est intéressant que les familles d'invariants disposent de certaines propriétés. Parmi elles, la *complétude* qui garantit la reconstruction de la forme à partir des descripteurs invariants ; et la *stabilité* qui assure la proximité des invariants lorsque les formes sont proches.

Une famille d'invariants $\{I_n\}$ sur $\mathcal{X}$ par rapport à $\mathcal{G}$ est dite complète, si et seulement si pour tout forme $\mathcal{F}$, on a :

$$O_1 \text{ et } O_2 \text{ ont la même forme } \mathcal{F} \iff I_n(\gamma_2) = I_n(\gamma_1), \quad \forall n \in \mathcal{J}$$

La propriété de complétude permet d'être assuré d'obtenir deux familles d'invariants différents pour deux formes différentes.

Une famille d'invariants $\{I_n\}$ est dite stable, si et seulement si ses descripteurs invariants définissent une fonction continue entre l'espace des paramétrisations et l'espace des descripteurs.

La propriété de stabilité se traduit par le fait qu'une faible variation de forme induit une faible variation sur la famille d'invariants correspondante. Cette propriété est importante car, de part la discrétisation des formes dans les images, une transformation de la forme résulte en une paramétrisation du contour plus ou moins différente. Afin d'atténuer ce problème, résultant de la discrétisation des images, il faut considérer l'*isotropie* de l'algorithme de représentation, qui a pour but l'interpolation de la courbe.

5.2.3.3 Signature du contour

Les descripteurs de Fourier sont calculés à partir du contour de la région de la main. Les points de ce contour peuvent être représentés avec différentes signatures (Mokadem [97]) :

- *cartésienne :* coordonnées complexes,
- *radiale :* distance au centre,
- *tangentielle :* courbure ou tangente au contour.

Zhang et Lu [142] ont comparé les performances de ces différentes signatures, et rapportent que la distance radiale donne les meilleurs résultats, et que la courbure donne les plus mauvais. Mais leurs tests sont effectués sur un nombre relativement faible de formes synthétiques.

C'est la signature complexe qui est généralement utilisée dans les travaux sur la reconnaissance de gestes (cf. section 5.1). La description *radiale* a l'inconvénient d'être limitée à des contours étoilés.

5.2.3.4 Transformée de FOURIER du contour

Nous utilisons les coordonnées complexes. Chaque point M_i du contour de la forme est représenté par un nombre complexe z_i, avec N le nombre de points du contour :

$$\forall i \in [0, N-1], \quad M_i(x_i, y_i) \Leftrightarrow z_i = x_i + jy_i \tag{5.16}$$

Avant de calculer la transformée de FOURIER du contour (avec la *Transformée de Fourier Rapide* (FFT [8])), le contour est échantillonné pour obtenir une longueur normalisée. Cet échantillonnage est réalisé en interpolant les points à égale distance en terme de longueur de l'arc. La longueur normalisée du contour est choisie de façon à trouver un compromis entre une bonne description de la forme, avec suffisamment de détails, et un lissage de la forme pour éliminer les plus petits détails qui sont assimilés à du bruit.

Dans notre base de données de gestes, le nombre de points du contour de la main varie de 90 à 742, en fonction de la position de la main face à la caméra. Un autre facteur important est le temps de calcul qui augmente avec le nombre de points. Pour l'optimiser, le nombre de points est choisi pour être un multiple de deux. Par conséquent, la longueur normalisée du contour est choisie à $N = 64$ points.

Après cette étape de normalisation, la transformée de Fourier appliquée à la signature complexe du contour donne N coefficients de Fourier, C_k :

$$C_k(\gamma) = \sum_{i=0}^{N-1} z_i \, e^{-j\frac{2\pi ik}{N}}, \quad k \in [0, N-1] \tag{5.17}$$

La figure 5.1 montre les spectres pour trois gestes extraits de notre base (présentée au paragraphe 2.5.3). Le geste 6, qui représente le poing fermé, est celui qui a l'énergie la plus faible, alors que le geste 11, qui représente la main ouverte avec les doigts écartés, a des coefficients plus élevés, dus aux doigts. Les autres gestes, comme par exemple le geste 2, ont un spectre qui se situe entre ceux des gestes 6 et 11.

La figure 5.2 montre le résultat de la reconstruction avec différents exemples de contours de la main, en fonction de la fréquence de coupure. Il apparaît clairement que les premiers coefficients de FOURIER, correspondant aux basses fréquences, contiennent l'information générale sur la forme du contour, le contour reconstruit est lissé ; alors que les coefficients plus élevés, correspondant aux hautes fréquences, contiennent les détails de la forme.

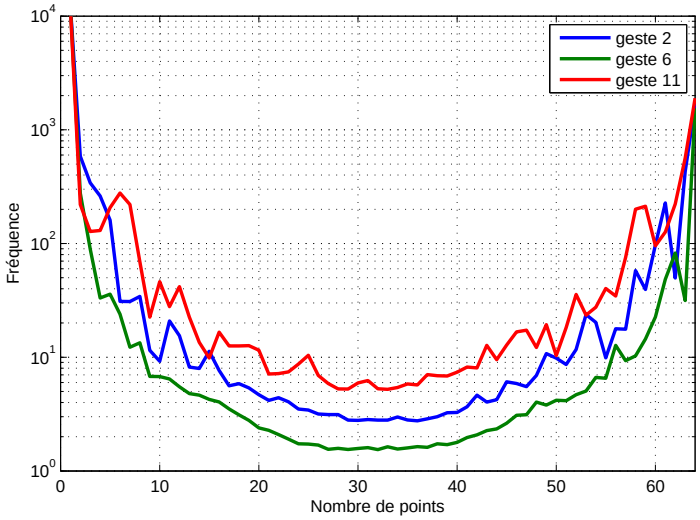
On constate sur cette figure, qu'à partir de vingt coefficients la forme du contour est bien reconstruite. Il n'est donc pas nécessaire de prendre en compte tous les coefficients pour la reconnaissance.

Reprenons l'équation 5.15 qui exprime la relation entre deux objets de contour fermé ayant la même forme et des représentations différentes, paramétrés par leurs abscisses curvilignes $\gamma_1(l)$ et $\gamma_2(l)$:

$$\gamma_2(l) = ae^{j\theta}\gamma_1(l + l_0) + b \tag{5.18}$$

8. *Fast Fourier Transform*

(a) *geste 2* (b) *geste 6* (c) *geste 11*



(d) *spectres*

FIGURE 5.1 – *Exemples de spectres de* FOURIER.

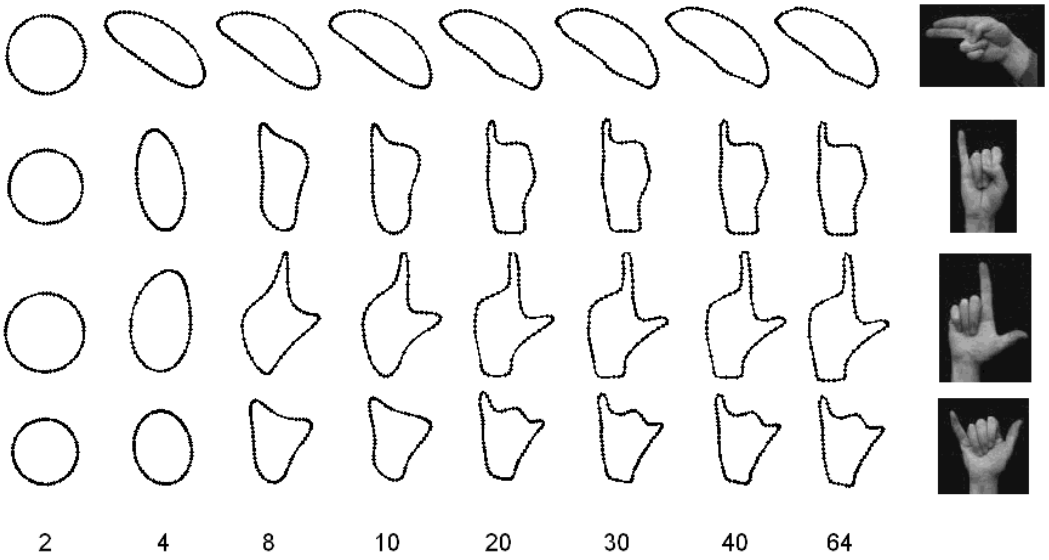


FIGURE 5.2 – *Exemple de reconstruction à partir des coefficients de* FOURIER*, en fonction de la fréquence de coupure, avec un contour initial échantillonné à 64 points.*

avec a un facteur d'échelle, θ l'angle de rotation, b le vecteur de translation, et l_0 la différence de point de départ entre les deux courbes.

Dans le domaine fréquentiel, en appliquant la transformée de FOURIER, on obtient :

$$C_k(\gamma_2) = ae^{j\theta}e^{j\frac{2\pi kl_0}{N}}C_k(\gamma_1) + b\delta_k \tag{5.19}$$

avec δ_k la fonction de KRONECKER. Le premier coefficient, C_0, contient la position de la forme. Il n'est donc pas pris en compte dans la suite. Une rotation de la forme et un changement de point de départ ne modifient que la phase.

5.2.3.5 Une famille d'invariants « commune » (FD1)

La méthode habituelle pour obtenir une famille invariante de descripteurs de FOURIER est de prendre le module des coefficients de FOURIER.

Nous obtenons ainsi $N-2$ descripteurs de FOURIER, notés I_k :

$$I_k = \frac{|C_k|}{|C_1|}, \quad k = 2, \ldots, N-1 \tag{5.20}$$

En prenant le module, l'équation 5.19 est simplifiée, ce qui permet les invariances aux rotations et au changement de point de départ, puisque ces transformations n'influent que sur la phase. L'invariance aux changements d'échelle est obtenue en divisant les coefficients par le module du second coefficient, C_1.

Cependant, cette famille d'invariants n'est pas complète, puisqu'elle ne contient pas l'information de phase de la forme.

5.2.3.6 Une famille d'invariants complète et stable (FD2)

Nous avons vu au paragraphe 5.2.3.2 que les propriétés de complétude et de stabilité sont très intéressantes pour une famille d'invariants. CRIMMINS [30] a proposé une famille d'invariants complète, mais qui n'est pas stable. Par le biais d'une normalisation judicieuse, cette famille a été rendue stable par GHORBEL [48] :

$$I_k = \frac{C_k^{k_0-k_1}C_{k_0}^{k_1-k}C_{k_1}^{k-k_0}}{|C_{k_0}^{k_1-k-p}||C_{k_1}^{k-k_0-q}|} \tag{5.21}$$

avec $(p,q) \in \mathbb{R}^+$, et $k_0 > k_1$.

Pour nos expérimentations, nous prendrons les paramètres suivants : $k_0 = 2$, $k_1 = 1$, $p = q = 0,5$, ce qui permet de simplifier l'équation précédente.

Pour s'assurer de l'invariance en translation, le contour est centré avec son centre de gravité. Pour l'invariance aux changements d'échelle, la longueur du contour est normalisée à 1.

5.3 CLASSIFICATION

La classification consiste à maximiser ou à minimiser une fonction discriminante $d_i(\mathbf{x})$ entre un vecteur de mesures $\mathbf{x}$ et les N classes de gestes. Par exemple, dans le cas d'une fonction à minimiser, telle qu'une distance, on cherche la classe C telle que :

$$C = \underset{i\in[1,N]}{\arg\min}\, d_i(\mathbf{x}) \tag{5.22}$$

Nous présentons les distances généralement utilisées : *euclidienne* et *bayésienne*, ainsi que l'étape d'apprentissage associée. Nous présentons ensuite la validation croisée, utilisée lorsque le nombre d'images disponibles est faible, ce qui est le cas de la base de TRIESCH. Enfin, nous présentons les classifieurs *k*-NN et SVM, que nous testons afin d'évaluer les éventuelles améliorations.

5.3.1 *Apprentissage*

L'apprentissage consiste à calculer le modèle de forme moyenne d'une classe de gestes, à partir d'un ensemble d'images d'apprentissage. Cette étape se fait généralement « hors-ligne ». En supposant que la variation entre les vecteurs caractéristiques des exemples d'images d'un même geste est de type gaussienne, nous calculons le vecteur moyenne μ_i et la matrice de covariance Λ_i pour chaque classe, $i \in [1, N]$, à partir des vecteurs d'invariants $\mathbf{x}_i^k$ des images d'apprentissage, avec $k \in [1, M_i]$ le nombre d'images d'apprentissage pour la classe i :

$$\mu_i = \frac{1}{M_i} \sum_{k=1}^{M_i} \mathbf{x}_i^k \tag{5.23}$$

$$\Lambda_i = \frac{1}{M_i} \sum_{k=1}^{M_i} (\mathbf{x}_i^k - \mu_i)(\mathbf{x}_i^k - \mu_i)^T \tag{5.24}$$

5.3.2 *Distance euclidienne*

La distance euclidienne entre le vecteur de mesure $\mathbf{x}$ et la classe i est définie par :

$$d_{E,i}(\mathbf{x}) = \sqrt{(\mathbf{x} - \mu_i)^T (\mathbf{x} - \mu_i)} \tag{5.25}$$

avec μ_i le vecteur moyenne de la classe i.

C'est la métrique usuelle pour calculer une distance entre les vecteurs d'invariants I_k de deux contours γ_1 et γ_2. Elle s'écrit dans ce cas :

$$d_E(\gamma_1, \gamma_2) = \sqrt{\sum_k |I_k(\gamma_1) - I_k(\gamma_2)|^2} \tag{5.26}$$

Par ailleurs, GHORBEL [48] a montré que l'ensemble d'invariants complet et stable (FD2) induit une distance dans l'espace des formes, qui est donnée par la distance euclidienne.

5.3.3 *Distance bayésienne*

La classification bayésienne est basée sur la règle de BAYES :

$$p(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)\, p(C_i)}{p(\mathbf{x})} \tag{5.27}$$

avec :

- $p(C_i \mid \mathbf{x})$ la probabilité *a posteriori* de la classe C_i, sachant que le vecteur de mesure est $\mathbf{x}$,
- $p(\mathbf{x} \mid C_i)$ la probabilité conditionnelle de $\mathbf{x}$, sachant que la classe est C_i,
- $p(C_i)$ la probabilité *a priori* de la classe C_i,
- $p(\mathbf{x})$ la probabilité du vecteur de mesure $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{i=1}^{N} p(\mathbf{x} \mid C_i)\, p(C_i) \tag{5.28}$$

Dans ce cas, la fonction discriminante est donnée par le maximum a posteriori :

$$d_i(\mathbf{x}) = p(C_i \mid \mathbf{x}) \tag{5.29}$$

La règle de BAYES peut se réécrire de la façon suivante (MARTIN [92]) :

$$d_i(\mathbf{x}) = d_{M,i}(\mathbf{x}) + log(|\Lambda_i|) \tag{5.30}$$

avec $d_M(\mathbf{x})$ la distance de MAHALANOBIS :

$$d_{M,i}(\mathbf{x}) = (\mathbf{x} - \mu_i)^T \Lambda_i^{-1} (\mathbf{x} - \mu_i) \tag{5.31}$$

Cette distance de MAHALANOBIS apparaît comme une distance euclidienne pondérée par les matrices de covariance de chaque classe.

5.3.4 *Validation croisée*

Le nombre d'images de la base de gestes de TRIESCH est faible, trop pour que les résultats de la classification soient fiables. Suivant la façon dont les images sont séparées entre ensembles d'apprentissage et de test, il est probable d'obtenir des résultats différents.

Lorsque, comme dans ce cas, l'ensemble E des N images est trop petit, il est possible d'effectuer une validation croisée. Il existe plusieurs méthodes de validation croisée, en fonction de la technique choisie pour déterminer les sous-ensembles d'images :

ALÉATOIRE : k échantillons sont tirés aléatoirement dans l'ensemble E pour chaque expérience. Le taux d'erreur est la moyenne des taux de chacune des expériences.

K-BLOCS [9] : l'ensemble E est divisé en K sous-ensembles disjoints de N/K échantillons chacun. Le test est fait avec un des sous-ensembles, et l'apprentissage avec les autres sous-ensembles. Le taux d'erreur est la moyenne des K expériences. L'avantage de cette méthode est que tous les échantillons de E sont utilisés.

N-BLOCS [10] : un seul échantillon est utilisé pour tester, ce qui revient à faire des blocs d'une image. Le taux d'erreur est la moyenne des N expériences. Cette méthode est utile lorsque le nombre d'images N est très faible.

Le choix de la valeur de K est fait en fonction de la taille de l'ensemble E :

- si le nombre d'images N est faible, K est grand afin d'utiliser au maximum les images de E pour l'apprentissage. Dans ce cas, la variance est plus grande et le biais plus petit ;
- si le nombre d'images N est plus grand, K est plus petit ce qui nécessite moins d'expériences. Dans ce cas, la variance est plus petite et le biais plus grand.

Pour la base de TRIESCH, nous utilisons une validation croisée de type « n-blocs » (blocs d'une image), car le nombre d'images est faible. Nous utilisons aussi la validation croisée sur notre base de gestes, avec des blocs de 50 images.

9. *K-fold cross-validation*
10. *Leave-one-out cross-validation*

5.3.5 *Autres classifieurs*

Après avoir testé les classifications euclidienne et bayésienne, nous allons comparer les résultats avec d'autres classifieurs couramment utilisés en reconnaissance de formes : les *k-plus proches voisins* (*k*-NN) et les SVM.

5.3.5.1 k-plus proches voisins

La méthode des *k-plus proches voisins* (*k*-NN [11]) est une des méthodes d'apprentissage les plus simples. Elle permet de classer un nouvel échantillon en faisant voter les k échantillons d'apprentissage les plus proches, dans l'espace des caractéristiques, avec une distance à définir (généralement la distance euclidienne).

5.3.5.2 SVM

Les « *Machines à Support Vectoriel* » (SVM [12]) [116] sont une classe d'algorithmes de classification, consistant à séparer deux (ou plusieurs) ensembles de points par une surface de décision (hyperplan). Les SVM sont basées sur l'utilisation de noyaux qui permettent une séparation optimale des points en plusieurs ensembles. La solution est optimale dans le sens où la marge, entre l'hyperplan et les vecteurs des données d'apprentissage de chaque classe, est maximale.

Les SVM permettent de résoudre le problème des données non séparables linéairement, en projetant les données dans un espace de dimension supérieure. Cette projection se fait avec un noyau polynomial, gaussien ou hyperbolique.

Hsu *et al.* [61] conseillent d'utiliser un noyau RBF [13] et expliquent comment choisir les paramètres, en testant différents jeux de paramètres avec une validation croisée.

5.4 RÉSULTATS ET INTERPRÉTATION

Cette section présente d'abord une comparaison des descripteurs avec la distance euclidienne, sur la base de Triesch et la nôtre (présentées dans la section 2.5). Nous comparons également les résultats en fonction des gestes et des personnes réalisant ces gestes. Nous comparons ensuite les résultats des classifieurs bayésien, *k*-NN et SVM.

La segmentation des images de la base de Triesch est réalisée avec la méthode de Otsu (cf. paragraphe 4.2.1). Les images de notre base sont segmentées avec les seuils C_bC_r (cf. paragraphe 4.2.3.1).

5.4.1 *Classification euclidienne*

5.4.1.1 Avec la base de gestes de Triesch

Les images de la base de Triesch sont divisées en deux sous-ensembles, l'un pour l'apprentissage et l'autre pour la classification. Cependant, le nombre d'images n'est pas suffisant pour que les résultats soient fiables. Pour résoudre

11. *k-Nearest Neighbors*
12. *Support Vector Machine*
13. *Radial-Basis Function*

	HU	ZERNIKE	FD1	FD2
Apprentissage	38,9	81,5	81,5	80,3
Validation croisée	37,1	74,9	77,8	77,0
Test	30,5	76,7	77,0	76,2

TABLEAU 5.1 – *Taux de reconnaissance (%) avec la base de* TRIESCH *et la distance euclidienne, avec 6 invariants pour les FD1, et 4 invariants pour les FD2.*

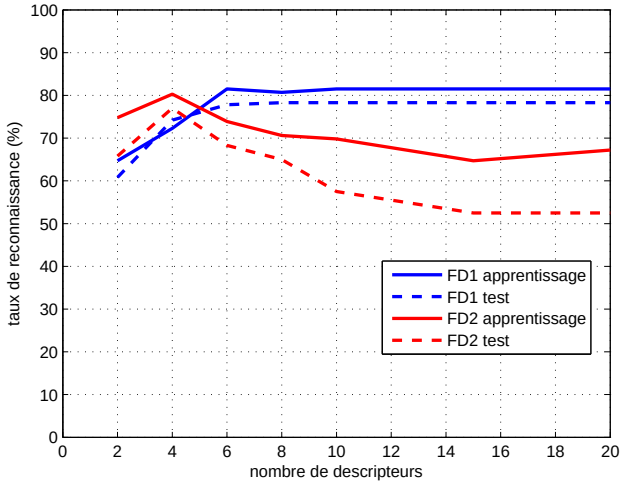


FIGURE 5.3 – *Taux de classification en fonction du nombre de descripteurs de* FOURIER, *pour la base de* TRIESCH, *avec les FD1 en bleu et les FD2 en rouge.*

ce problème, nous effectuons également une validation croisée de type « n-blocs » (blocs d'une image) sur l'ensemble des images.

Le tableau 5.1 montre que les moments de HU donnent un taux de reconnaissance faible (37,1% avec la validation croisée), alors qu'avec les autres techniques les résultats sont meilleurs.

Pour les descripteurs de FOURIER, le taux de reconnaissance dépend du nombre de descripteurs utilisés pour calculer la distance. La figure 5.3 montre qu'il est suffisant de prendre 6 descripteurs pour les FD1, et 4 descripteurs pour les FD2. Cette figure montre aussi que, pour les FD2, le taux de reconnaissance diminue lorsque le nombre de descripteurs augmente. Ceci peut s'expliquer par une sensibilité plus importante de cette famille, qui est plus facilement perturbée par la présence de détails sur le contour, s'assimilant à du bruit.

Les résultats des moments de ZERNIKE sont similaires aux FD, mais leur temps de calcul avec la méthode directe est très élevé : 1,20 secondes/image environ contre 10 millisecondes/image pour les FD et les moments de HU. Toutefois, nous avons évoqué au paragraphe 5.2.2 des méthodes qui permettrait de diminuer fortement ce temps de calcul.

Pour vérifier les invariances, nous avons effectué des tests avec des images ayant subi une rotation et un changement d'échelle. Avec des angles de rotation de 90°, 180° et 270°, nous obtenons les mêmes résultats avec les descripteurs de FOURIER que pour les images non transformées. Toutefois, pour des angles de rotation qui ne sont pas multiples de 90°, par exemple 30° et 60°, les taux de

	HU	ZERNIKE	FD1	FD2
Apprentissage	62,9	67,6	95,7	86,3
Validation croisée	60,6	66,9	95,3	78,6
Test #1	42,7	30,5	78,7	49,2
Test #2	46,1	53,9	80,2	18,2
Test #3	34,2	26,0	71,7	61,2
Test #4	33,1	18,9	73,1	52,1
Test total	39,2	32,5	76,0	45,1

TABLEAU 5.2 – *Taux de reconnaissance (%) avec notre base et la distance euclidienne, avec 6 invariants pour les FD1, et 4 invariants pour les FD2. La validation croisée est réalisée sur l'ensemble d'apprentissage par blocs de 50 images. Les lignes « Test #i » correspondent aux résultats de 4 personnes séparément.*

reconnaissance diminuent. Les erreurs sont dues à l'interpolation des formes après leur rotation, pour que les pixels du contour correspondent à la grille discrète des images.

5.4.1.2 Avec notre base de gestes

L'apprentissage est réalisé avec les images d'un utilisateur « expert », qui a réalisé les gestes de la façon la plus conforme possible au vocabulaire, là où certains utilisateurs « non initiés » ne plieraient les doigts qu'à moitié au lieu de les replier complètement. Environ 500 images ont été sélectionnées manuellement pour chaque geste, afin d'avoir des images bien représentatives. Dans les tests suivants nous utilisons 6 invariants pour les FD1 et 4 pour les FD2.

Pour valider la phase d'apprentissage, nous effectuons une classification des images d'apprentissage. Nous procédons également à une validation croisée sur l'ensemble d'apprentissage, avec des blocs de 50 images. Ensuite, les images des autres utilisateurs sont utilisées comme ensemble de test, avec environ 1 000 images par geste et par personne. La figure 2.9 montre des exemples d'images utilisées.

Le tableau 5.2 montre que les meilleurs taux de reconnaissance sont obtenus avec les FD1. Les FD2 donnent de bons résultats sur l'ensemble d'apprentissage, mais de moins bons résultats pour l'ensemble de test. Ce tableau présente aussi les résultats de 4 personnes (Tests #1 à #4), ce qui met en exergue le fait que le taux de reconnaissance est très variable d'une personne à l'autre, suivant la manière dont les gestes sont réalisés.

Le tableau 5.3 montre les taux de reconnaissance de chaque geste avec l'ensemble de test. On constate notamment que les gestes 2, 3 et 8 sont très mal classifiés avec les moments de HU et les moments de ZERNIKE. Les gestes 7 et 11 donnent aussi de mauvais résultats, y compris avec les FD2.

L'analyse de la matrice de confusion pour les FD1 (tableau 5.4) montre quels sont les gestes qui causent le plus d'erreurs. Par exemple, les gestes 2, 3 et 9, très similaires, sont confondus avec les autres méthodes, alors qu'ils sont bien reconnus avec les FD1. Le geste 8 est confondu avec le geste 9, résultat qui n'est

	1	2	3	4	5	6	7	8	9	10	11	TOTAL
HU	83,2	10,6	6,9	47,8	61,7	100	26,3	0,9	52,9	37,1	28,1	42,7
ZER.	72,9	1,5	2,1	22,9	82,4	90,5	6,2	1,1	6,6	9,1	11,2	30,5
FD1	93,6	56,7	75,2	96,6	97,1	100	86,1	30,7	87,2	74,9	63,9	78,7
FD2	40,3	82,4	56,6	52,6	73,0	53,1	16,5	16,4	46,9	81,5	22,7	49,2

TABLEAU 5.3 – *Taux de reconnaissance (%) pour chaque geste, avec la personne #1, pour notre base et la distance euclidienne.*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	**93,6**	0,0	0,0	1,6	1,2	0,0	0,0	0,0	0,0	0,0	0,0
2	0,2	**56,7**	0,1	0,0	0,1	0,0	0,0	0,0	0,0	1,7	0,0
3	0,0	23,0	**75,2**	0,0	0,0	0,0	0,0	0,0	0,0	4,4	11,6
4	3,5	0,0	0,0	**96,6**	1,6	0,0	0,0	0,0	0,0	0,0	0,0
5	1,9	1,5	1,7	1,8	**97,1**	0,0	0,0	0,0	0,0	3,8	0,0
6	0,9	0,0	0,0	0,0	0,0	**100**	0,0	0,0	0,0	6,1	2,0
7	0,0	0,1	7,3	0,0	0,0	0,0	**86,1**	4,8	2,7	7,0	21,9
8	0,0	0,0	0,1	0,0	0,0	0,0	1,7	**30,6**	10,1	2,0	0,0
9	0,0	18,7	1,1	0,0	0,0	0,0	3,4	57,8	**87,2**	0,0	0,0
10	0,0	0,0	1,3	0,0	0,0	0,0	8,8	6,7	0,0	**74,9**	0,6
11	0,0	0,0	13,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	**63,9**

TABLEAU 5.4 – *Matrice de confusion pour les FD1*

pas vraiment étonnant, puisque ces gestes ont été choisis pour tester les limites de la reconnaissance. Par contre, les gestes 4 et 5 sont bien différenciés.

On constate également sur la matrice de confusion que le geste 11 est parfois confondu avec les gestes 3 et 7. Dans les deux cas, il s'agit d'une mauvaise segmentation qui ne permet pas de séparer correctement les doigts, ou de tous les détecter.

5.4.1.3 Synthèse

Les différentes expérimentations avec la distance euclidienne démontrent que la première famille de descripteurs (FD1) permet d'obtenir les meilleurs résultats sur les deux bases évaluées : 76% sur l'ensemble de test de notre base, et 95,3% avec la validation croisée. Les FD1 permettent de discriminer des gestes qui sont très proches visuellement, contrairement aux autres descripteurs.

La deuxième famille (FD2) ne donne pas les résultats espérés. Les moins bonnes performances des FD2 s'expliquent par une plus grande sensibilité au bruit, et par des problèmes de stabilité numérique. En effet, cette famille utilise la phase, qui contient beaucoup d'information, mais qui est difficile à exploiter.

Pour améliorer les résultats, deux solutions sont suggérées : la première consiste à améliorer la segmentation, et la deuxième à ajouter des images de l'utilisateur dans l'ensemble d'apprentissage. Cette solution se justifie par les résultats de validation croisée, pour laquelle l'apprentissage et la classifica-

	HU	ZERNIKE	FD1
Apprentissage	73,2	99,2	100,0
Validation croisée	67,8	76,5	81,6
Test	55,0	54,2	75,3

TABLEAU 5.5 – *Taux de reconnaissance (%) avec la base de* TRIESCH *et la distance bayésienne, avec 20 invariants pour les* FD1.

tion sont fait sur les images d'une seule personne, alors que les ensembles d'apprentissage et des tests n'utilisent pas les mêmes images.

5.4.2 *Classification bayésienne*

Au vu des résultats précédents, avec la classification euclidienne, il apparaît que les FD1 donnent les meilleurs taux de reconnaissance. Nous allons donc concentrer nos tests sur cette famille, tout en la comparant avec les moments de HU. Les moments de ZERNIKE n'ont été que partiellement testés avec la classification bayésienne, à cause de leur temps de calcul trop élevé. L'apprentissage et la classification sont réalisés de la même façon qu'à la section précédente.

5.4.2.1 Avec la base de gestes de TRIESCH

Pour les descripteurs de FOURIER, le taux de reconnaissance dépend du nombre de descripteurs (figure 5.4) : avec le même nombre de FD qu'au paragraphe précédent (6), le taux est d'environ 95% pour l'ensemble de test. On constate qu'avec 13 FD, il atteint 100% pour l'apprentissage et 80% pour la validation croisée. Le tableau 5.5 montre qu'en prenant toutes les images, à la fois pour l'apprentissage et pour les tests, nous obtenons un taux de reconnaissance de 73,2% pour les moments de HU, avec des taux faibles pour les gestes « b » et « g » de la figure 2.8 (tableau non reproduit). Les moments de ZERNIKE donnent de bons résultats pour l'apprentissage et la validation croisée, mais le taux de reconnaissance est plus faible pour les tests (54,2%).

Par ailleurs, les résultats sont meilleurs avec la validation croisée qu'avec l'ensemble de test. Cela peut s'expliquer par le fait que le nombre d'images de test est trop faible puisqu'on divise dans ce cas les images en deux sous-ensembles, pour l'apprentissage et les tests.

5.4.2.2 Avec notre base de gestes

Nous avons vu au paragraphe précédent que le nombre de descripteurs utilisés pour la classification joue un rôle important. Nous comparons donc les résultats avec 6 et 20 invariants pour les FD1, et avec 6 et 7 invariants de HU. En effet, le septième invariant de HU permet de distinguer des images symétriques, par exemple la main gauche et la main droite lorsque le pouce est présent. Or, dans notre base de gestes, les utilisateurs ont utilisé indifféremment leur main gauche ou leur main droite. Ainsi, en ne prenant pas en compte le septième invariant de HU, le système doit pouvoir reconnaître une main droite si l'apprentissage a été effectué avec une main gauche, et inversement.

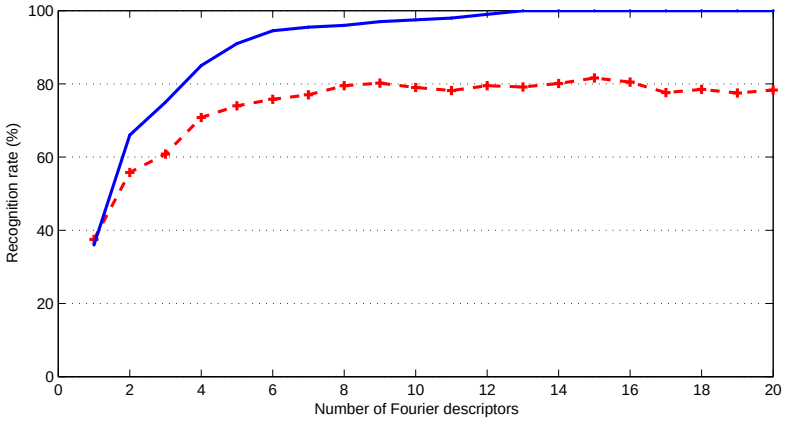


FIGURE 5.4 – *Taux de classification en fonction du nombre de descripteurs de* FOURIER, *pour la base de* TRIESCH, *avec la classification des images d'apprentissage en bleu et la validation croisée en rouge pointillés.*

	6 HU	7 HU	6 FD1	20 FD1
Apprentissage	98,43	99,13	99,91	99,96
Test	71,08	59,07	84,58	91,30

TABLEAU 5.6 – *Taux de reconnaissance (%) avec notre base et la distance bayésienne, avec 6 ou 7 moments de* HU, *et 6 ou 20 descripteurs pour les FD1.*

Pour valider la phase d'apprentissage, nous effectuons une classification des images d'apprentissage. Le tableau 5.6 montre que, quelque soit la méthode et le nombre d'invariants, le résultat est très satisfaisant (98-99%).

Ensuite, les images des autres personnes sont classifiées, avec environ 1 000 images par geste et par personne. On constate qu'en ne prenant pas le septième invariant de HU, le taux de reconnaissance augmente de 59% à 71%. Ce tableau confirme aussi que le résultat est meilleur avec 20 descripteurs de FOURIER dans le cas de classification bayésienne.

Le tableau 5.7 montre les résultats de classification pour chaque geste, avec 6 FD et 6 invariants de HU. La différence entre les deux méthodes s'explique par une amélioration significative de la reconnaissance des gestes 2, 3, 7 et 11.

L'analyse des matrices de confusion (tableaux 5.8 et 5.9) confirme les résultats et révèle les gestes qui sont sources de confusion. Par exemple, les gestes 2, 3 et 9 sont très similaires et sont confondus par les moments de HU, alors qu'ils sont bien classifiés avec les FD. Par contre, pour les gestes 4 et 8, le taux de reconnaissance n'est pas amélioré, ce qui n'est pas réellement surprenant puisque ces gestes ont été choisis pour tester les limites de la classification. Si nous retirons ces deux gestes et que nous réalisons à nouveau l'apprentissage et la classification, le taux de reconnaissance passe de 84,58% à 90,52% avec 6 FD1, et de 71,08% à 72,97% avec 6 moments de HU.

5.4.2.3 Synthèse

Comme pour la classification euclidienne, les FD1 donnent les meilleurs taux de reconnaissance avec la classification bayésienne. Par ailleurs, l'amélioration

	1	2	3	4	5	6	7	8	9	10	11	TOTAL
6 HU	82,3	68,3	66,5	65,0	93,8	89,0	48,3	65,4	82,8	74,4	40,6	**71,08**
6 FD1	86,6	90,8	96,4	60,8	97,8	94,3	80,6	64,8	88,6	73,4	96,2	**84,58**

TABLEAU 5.7 – *Taux de reconnaissance (%) par geste, avec notre base et la distance bayésienne, avec 6 moments de* HU *et 6 descripteurs pour les FD1.*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	**82,3**	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	4,7	**68,3**	1,5	1,2	0,7	0,6	1,2	0,5	0,1	0,0	2,3
3	0,7	21,5	**66,5**	0,5	1,1	2,5	20,0	20,3	10,2	2,0	40,1
4	3,5	0,2	0,0	**65,0**	2,0	0,0	0,5	0,1	0,0	0,7	0,0
5	5,6	8,3	0,8	31,2	**93,8**	0,7	6,2	4,2	1,6	11,7	0,0
6	0,8	0,0	0,0	0,1	0,2	**89,0**	0,0	0,0	0,0	0,3	0,0
7	0,2	0,1	0,4	0,2	0,2	0,6	**48,3**	1,0	0,7	0,3	0,6
8	1,9	0,4	1,8	1,2	1,4	0,4	4,4	**65,4**	3,7	1,2	0,2
9	0,0	1,0	28,8	0,1	0,1	0,5	15,2	8,1	**82,8**	7,3	16,2
10	0,0	0,1	0,0	0,2	0,1	0,2	0,2	0,0	0,0	**74,4**	0,0
11	0,2	0,2	0,1	0,3	0,2	5,6	4,0	0,5	1,0	2,0	**40,6**

TABLEAU 5.8 – *Matrice de confusion pour les moments de* HU *(avec 6 invariants).*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	**86,6**	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	0,0	**90,8**	0,4	0,4	0,2	0,2	0,1	0,0	1,7	0,1	0,1
3	0,0	0,7	**96,4**	0,5	0,4	1,0	0,4	0,0	0,7	0,1	3,3
4	5,5	0,0	0,0	**60,8**	0,0	0,1	0,4	0,0	0,0	0,0	0,0
5	2,9	1,8	0,5	35,9	**97,8**	0,9	7,8	3,2	4,9	20,2	0,1
6	4,6	0,1	0,0	0,1	0,3	**94,3**	0,8	0,0	0,2	2,0	0,0
7	0,2	0,4	0,1	0,7	0,5	1,1	**80,6**	8,3	0,3	2,8	0,0
8	0,0	0,2	0,0	0,3	0,3	0,1	1,9	**64,8**	2,8	0,5	0,0
9	0,0	5,9	1,7	0,9	0,3	0,4	6,0	23,2	**88,6**	0,9	0,4
10	0,0	0,1	0,1	0,3	0,0	0,2	0,8	0,4	0,2	**73,4**	0,0
11	0,2	0,2	0,8	0,1	0,1	1,6	1,1	0,1	0,7	0,0	**96,2**

TABLEAU 5.9 – *Matrice de confusion pour les FD1 (avec 6 descripteurs).*

	BAYES	SVM	k-NN	EUCL.
TRIESCH, Toutes les images	100,0	89,1	93,3	77,0
Notre base, Apprentissage	99,9	99,9	100,0	95,7
Notre base, Test avec 6 FD1	84,6	84,2	87,9	76,0
Notre base, Test avec 20 FD1	91,3	87,9	89,2	-

TABLEAU 5.10 – *Taux de reconnaissance (%) avec les FD1 et différents classifieurs : bayésien, « Support Vector Machine » (SVM), k-plus proches voisins (k-NN) et distance euclidienne (Eucl.)*

des résultats avec les moments de HU est significative. Ceci montre que la distance euclidienne n'est pas adaptée aux moments de HU, ce qui peut s'expliquer par le fait que les moments sont d'ordre différent (ordres 2 et 3), et n'ont donc pas le même « poids ». Pour les descripteurs de FOURIER, on constate que les résultats sont très bons avec 6 descripteurs et encore meilleurs avec 20.

5.4.3 *Tests avec différents classifieurs*

Dans les sections précédentes, nous avons comparé les descripteurs avec la distance euclidienne. Le fait que les FD1 donnent les meilleurs résultats a été confirmé avec la distance bayésienne. Toutefois, il existe d'autres classifieurs plus évolués. Dans ce paragraphe, nous évaluons donc les classifieurs k-NN et SVM (cf. paragraphe 5.3.5) avec les FD1, afin de voir si les résultats sont améliorés.

Le tableau 5.10 montre que, pour la base de TRIESCH, le taux de reconnaissance est meilleur avec la classification bayésienne, avec un taux de 100%. Pour notre base de gestes, les résultats sont assez proches pour les trois classifieurs, avec un léger avantage pour la classification bayésienne.

5.5 AMÉLIORATION DE LA RECONNAISSANCE

Après avoir étudié la reconnaissance de postures avec des bases de données d'images, afin de comparer les descripteurs et les classifieurs, nous nous intéressons à la reconnaissance dans un flux vidéo. Nous présentons deux méthodes pour améliorer les résultats de classification. La première méthode permet de tirer partie de l'information temporelle, en supposant que lors de la réalisation d'un geste, la main reste dans la même configuration pendant un certain laps de temps.

La deuxième méthode vise à rejeter les gestes considérés comme trop différents de ceux du vocabulaire, que ce soit des gestes « inconnus », ou des gestes « ambigus » pour lesquels la décision entre deux gestes est difficile. En effet, lors de la réalisation de gestes dans un flux vidéo, la configuration de la main passe par des gestes de transition entre deux gestes du vocabulaire.

5.5.1 *Filtrage temporel*

Lors de la réalisation d'un geste dans un flux vidéo, il est raisonnable de supposer que le geste réalisé est le même pendant quelques secondes. Il est donc possible de filtrer temporellement les résultats de la classification afin de diminuer le taux d'erreur.

Nous proposons de moyenner la distance entre le geste à classer et chaque classe, avec une fenêtre glissante sur les N dernières images. En notant $d_i(k)$ la distance entre le geste à classer à l'image k et la classe i, la moyenne $\mu_i(k)$ de la distance à la classe i est donnée par :

$$\mu_i(k) = (1-\alpha)\mu_i(k-1) + \alpha\, d_i(k) \tag{5.32}$$

où $\alpha = 1/N$ est le facteur d'oubli. Les distances $\{d_i(k)\}_{i\in[1,M]}$ sont normalisées à 1 avant de calculer la moyenne.

Dans un deuxième temps, il peut être intéressant de prendre en compte le nombre de fois qu'un geste est reconnu dans les dernières images. Pour réaliser ceci, nous associons un poids $w_i(k)$ à chaque classe i. Les poids sont mis à jours de la façon suivante :

$$w_i(k) = (1-\alpha)w_i(k-1) + \alpha\, M_i(k) \tag{5.33}$$

où $M_i(k) = 1$ pour le geste qui a été reconnu, et 0 pour les autres. Enfin, on normalise les poids en divisant chaque $w_i(k)$ par leur somme. Le geste reconnu est alors celui dont le poids est le plus élevé.

5.5.2 *Méthode de rejet*

Une source importante d'erreur de classification est la présence de gestes qui ne correspondent pas à ceux du vocabulaire. Ceci peut arriver par exemple lors de la transition entre deux gestes, ou lorsque la main est mal segmentée. Les erreurs de segmentation altèrent fortement la forme de la main, par exemple en « collant » les doigts entre eux. Il est préférable dans ce genre de situation de rejeter les gestes en question plutôt que de les classer avec un risque d'erreur élevé.

Pour détecter les gestes « inconnus », ou ceux de transitions, nous avons étudié les distances aux classes $d_i(k), i \in [1, M]$ pour des gestes bien et mal classifiés. La première solution intuitive est de seuiller la distance, les distances étant normalisées par la distance maximale. Avec une distance inférieure au seuil, le geste est classifié, sinon il est rejeté. Mais il est toujours délicat de déterminer un seuil fixe qui donne de bons résultats dans toutes les situations.

La solution que nous proposons est de seuiller l'écart relatif entre les deux classes ayant les distances les plus petites. Cette solution repose sur l'idée que pour un geste mal segmenté ou pour un geste de transition, l'écart entre les distances des deux classes les plus proches est faible. De plus, ces distances sont élevées car le geste à reconnaître ne correspond pas à de ceux du vocabulaire. En triant les distances aux classes dans un ordre ascendant, cette distance β s'écrit :

$$\beta = \frac{d_1 - d_0}{d_0} \tag{5.34}$$

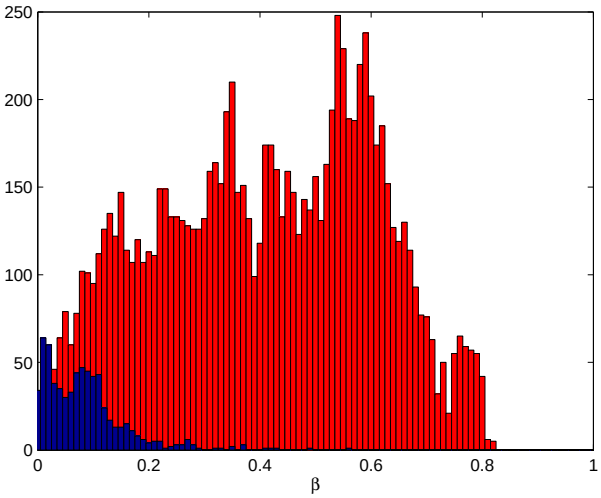


FIGURE 5.5 – *Histogramme de la distance β, avec les gestes bien classifiés en rouge, et mal classifiés en bleu.*

où d_0 est la distance la plus faible, celle de la classe qui devrait a priori correspondre au geste, et d_1 la deuxième distance la plus faible.

Ainsi, la mesure β donne une mesure de confiance sur la classification (figure 5.5) : si elle est élevée, alors $d_0 \gg d_1$, et il est très probable que le geste soit bien classifié. Mais si elle est faible, alors $d_0 \simeq d_1$, il existe une incertitude sur la classification.

5.5.3 *Résultats*

Nous avons testé le filtrage temporel et la méthode de rejet avec notre base de gestes et la classification bayésienne. Pour le filtrage temporel, avec la moyenne des distances sur 10 images ($\alpha = 0,1$) le taux de reconnaissance augmente de 91,3% à 93,77%. En rejetant les gestes pour lesquels la distance est supérieure à $0,4$, on obtient un taux de 94,16%. La méthode des poids donne un taux de reconnaissance de 93,86%. La combinaison de ces différentes méthodes permet d'atteindre 94,76% de gestes reconnus.

Pour la méthode de rejet, le tableau 5.11 donne le taux de rejet (pourcentage d'images rejetées) et le taux de reconnaissance sans les images rejetées, avec différentes valeurs de seuil appliquées à la mesure β. Plus β est élevé, plus le taux de reconnaissance augmente, au prix d'un taux de rejet plus important. Il faut aussi noter que pour certains gestes le taux de rejet est plus important que pour d'autres, en l'occurrence les gestes 2 et 10, et surtout le geste 11.

Ainsi, ces deux méthodes améliorent fortement le taux de reconnaissance, avec en contrepartie un certain nombre d'images non utilisées pour la classification. Pour un système de reconnaissance de gestes dans un flux vidéo, avec typiquement 20 à 25 images par seconde, rejeter une partie des images ne présente pas forcément un inconvénient à partir du moment où il y a suffisamment d'images et où cela permet de diminuer le taux d'erreur.

	$\beta = 0,05$	$\beta = 0,1$	$\beta = 0,15$	$\beta = 0,2$	$\beta = 0,25$
Rejet (%)	5,72	12,21	18,54	25,31	32,49
Reco. (%)	95,56	97,74	98,84	99,56	99,75

TABLEAU 5.11 – *Taux de reconnaissance et de rejet (%) avec la méthode de détection de gestes « inconnus », avec notre base de gestes et la classification bayésienne, avec 20 FD1.*

5.5.4 *Utilisation de deux caméras*

Étant donné que nous disposons de deux vues de la scène, il est possible de tirer partie de cette information supplémentaire pour améliorer la reconnaissance. En effet, le point de vue des deux caméras est légèrement différent. Par conséquent, suivant l'orientation de la main, il est probable qu'une des deux caméras permette une reconnaissance plus fiable que l'autre. Pour déterminer celle des deux caméras qui fournit la reconnaissance la plus fiable, nous utilisons la distance issue de la classification : le geste reconnu est celui de la caméra pour laquelle la distance est la plus faible.

Nous avons évalué l'apport de cette approche sur les séquences stéréoscopiques « `divers_L_1.avi` » et « `divers_R_1.avi` » (déjà utilisées pour comparer les méthodes de détection, cf. paragraphe 2.5.1 pour le détail des postures réalisées dans la séquence). Dans ces séquences, la main décrit cinq des postures de notre vocabulaire. Toutefois, ces séquences ont été acquises pour tester le suivi 3D de la main et non la reconnaissance de postures : le bras est nu, ce qui nécessite une détection du poignet avec la méthode présentée au paragraphe 4.3.2, et la forme de la main est très variable (cf. figure 6.10). De plus, cette séquence comprend les transitions entre les gestes.

Si nous réalisons la classification indépendamment sur chaque vue, avec la distance bayésienne et 20 FD1, les taux de reconnaissance sont de 56,5% et 59,2%. Avec la fusion des deux vues, le taux est de 68,6%, ce qui représente une amélioration de plus de 15%. De plus, en utilisant les méthodes de filtrage et de rejet présentées dans les paragraphes précédents, le taux de classification sur chaque vue est de l'ordre de 69%, et en fusionnant les deux vues il atteint 79,5%.

Ainsi, on constate que l'information supplémentaire fournie par une deuxième vue de la scène peut être utilisée pour améliorer la reconnaissance de postures, en fusionnant les résultats issus de chaque vue.

5.6 RÉSUMÉ

Dans ce chapitre, nous avons présenté la reconnaissance de gestes de la main à partir de descripteurs de formes [27], permettant d'obtenir un vecteur de caractéristiques invariantes : les moments de HU, les moments de ZERNIKE et les deux familles de descripteurs de FOURIER (FD). Nous avons utilisé deux bases de données de gestes : celle de TRIESCH [126], référence dans le domaine, et notre propre base de gestes, acquise dans le but de tester les invariances en rotation, translation et changement d'échelle des descripteurs, et d'obtenir des résultats représentatifs, correspondant aux conditions de notre application. Notre base

repose sur un vocabulaire de 11 gestes, réalisés par 18 personnes, avec environ 1 000 images par geste et par personne.

Nous avons d'abord évalué les résultats avec la classification euclidienne. La deuxième famille de FD (FD2) donne de moins bons résultats que la première (FD1), et ce malgré ses propriétés de complétude et stabilité (76% pour les FD1 contre 45,1% pour les FD2, pour notre base). Ceci s'explique par un écart entre la théorie et la réalité, dû à une plus grande sensibilité au bruit, et à des problèmes de stabilité numérique. Les moments de ZERNIKE obtiennent des résultats moyens et nécessitent un temps de calcul très élevé. Les moments de HU présentent aussi des performances moyennes. La classification bayésienne améliore fortement les résultats : les FD1 donnent les meilleurs taux de reconnaissance, avec 91,30% pour notre base. L'amélioration des résultats avec les moments de HU est aussi significative (71,08%).

Nos expérimentations montrent que la classification est très dépendante de la forme de la main (et donc de la segmentation) et de l'utilisateur. La façon dont les gestes sont réalisés est très variable suivant les personnes. Une solution pour améliorer les résultats est donc d'ajouter des images de l'utilisateur dans l'ensemble d'apprentissage.

Enfin, nous avons proposé trois méthodes pour améliorer les résultats dans le cas du traitement d'un flux vidéo. La première permet de tirer partie de l'information temporelle, en supposant qu'un même geste est réalisé pendant quelques secondes. La deuxième permet de détecter les gestes « inconnus » ou « ambigus », c'est-à-dire les gestes qui ne correspondent pas à un des gestes du vocabulaire, ce qui se produit typiquement lors des transitions. Les deux méthodes de filtrage permettent d'améliorer significativement le taux de reconnaissance, qui atteint 97,7% avec 12,2% de rejet, et 99,75% avec 32,5% de rejet. On obtient un taux de reconnaissance très élevé, au prix d'un taux de rejet important, résultat qui n'est pas gênant dans le cas du traitement d'un flux vidéo puisque le nombre d'images est suffisant. De plus, pour une application, il est préférable d'être certain de reconnaître le bon geste, et donc d'avoir un taux de fausses alarmes très faible.

La troisième méthode consiste à fusionner les résultats de classification issus de deux vues, dans le cadre de la vision stéréoscopique, afin de garder le meilleur des deux classifications. Si l'amélioration n'est pas négligeable, il est possible de mieux exploiter l'information stéréoscopique pour retrouver la configuration 3D de la main. Il s'agit de l'objectif du chapitre suivant.

6

SUIVI TRIDIMENSIONNEL DE LA MAIN

Dans ce chapitre, nous nous intéressons au suivi tridimensionnel du mouvement des doigts et de la main, en temps réel. Nous utilisons la vision stéréoscopique avec deux caméras vidéo calibrées, dans lesquelles la main est segmentée avec la méthode présentée dans le paragraphe 4.2.6. Les matrices de projection des caméras (paramètres intrinsèques et extrinsèques) sont connues grâce à une étape de calibration. Les principes généraux de la vision stéréoscopique, que nous utilisons dans ce chapitre, sont présentés en annexe B, page 125.

NB : pour parler des deux caméras, nous employons fréquemment les termes « image gauche » et « image droite »

Le suivi temporel permet d'améliorer la robustesse de la détection des points caractéristiques, et d'éliminer les fausses détections. L'objectif est aussi de connaître la position de chaque doigt au fil du temps, en associant chaque détection au doigt correspondant.

Après une introduction sur le domaine du suivi de la main, nous présentons plusieurs méthodes de suivi :

SUIVI DES DOIGTS EN 3D : un filtre de KALMAN permet d'estimer la position 3D de chaque doigt avec des observations bruitées. Cette méthode est d'abord appliquée au geste de pointage, avant d'être étendue au suivi multi-doigt.

SUIVI 2D DE LA MAIN : nous définissons un modèle squelettique simple de la main, prenant en compte les degrés de liberté principaux de la main. Ce modèle est recalé dans les images avec des points caractéristiques tels que le centre de la main et les bouts des doigts.

SUIVI 3D DE LA MAIN : l'approche par modèle squelettique est étendue en 3D, en combinant le modèle estimé dans les deux vues et en visualisant le résultat avec un modèle 3D. Ce dernier permet également de prendre en compte des contraintes supplémentaires sur la morphologie de la main.

SOMMAIRE

6.1 INTRODUCTION

Le suivi de mouvement en vision par ordinateur consiste à estimer la position et le mouvement d'objets ou d'êtres humains dans des séquences vidéos, avec une ou plusieurs caméras (BLACK ET ELLIS [7]). Le choix de la méthode de suivi dépend en partie de la précision voulue, si des erreurs sur le calcul de la position sont acceptables ou non, et dans quelle mesure. En effet, suivant l'application, la précision nécessaire ne sera pas la même.

De plus, selon LACHENAL [84], les mouvements non contraints des doigts d'une personne se font à une vitesse moyenne de 2 m/s et peuvent atteindre 5 m/s, avec de fortes variations d'accélération, dans les conditions de leurs expérimentations. Les mouvements brusques ou erratiques des doigts sont donc difficiles à appréhender dans un cadre de vision par ordinateur.

Dans notre cas, l'objectif du suivi de la main et des doigts dans une séquence vidéo est de rendre plus robuste la détection en tirant avantage de l'information temporelle et de la redondance d'information entre les images successives. De plus, le suivi permet de diminuer le temps de calcul, en n'effectuant qu'une mise à jour des paramètres au lieu de les recalculer à chaque image.

L'objectif est aussi de connaître la position de chaque doigt au fil du temps. En effet, pour l'instant nous sommes en mesure de détecter les doigts et d'obtenir leurs positions dans une image, mais nous ne pouvons pas calculer la trajectoire de chaque doigt. Pour ce faire, il faut associer à chaque doigt la détection correspondante.

Les applications de la reconnaissance de gestes telles que les surfaces interactives présentées à la paragraphe 3.1.2, montrent l'intérêt du suivi d'un ou plusieurs doigts pour la réalité augmentée. Par exemple, le système `DigitalDesk`, de CROWLEY *et al.* [31], permet de suivre un doigt en 2D sur une surface. Le suivi est effectué par corrélation avec un modèle de bout de doigt. Les trajectoires obtenues ont été utilisées par MARTIN ET DURAND [93] pour la reconnaissance d'écriture 2D avec des *Modèles de Markov Cachés* (HMM [1]).

Le système `EnhancedDesk` [100] permet de suivre plusieurs doigts en 2D, avec un filtre de KALMAN pour chaque doigt. Les bouts de doigts sont détectés grâce à une caméra infrarouge et une mesure de corrélation. Des gestes symboliques (cercle, carré, triangle, ...) sont ensuite reconnus avec des HMM.

Une application classique dans ce domaine est le remplacement de la souris par la main. La position du curseur sur l'écran fournit un retour d'information à l'utilisateur, validant ainsi la bonne localisation du doigt. Par exemple, SEGEN ET KUMAR [114] utilisent des points de contours caractéristiques, détectés avec la courbure du contour, pour classer quatre gestes de la main. Ils déterminent ensuite la direction 3D pointée par le doigt, avec deux caméras. HUNG *et al.* [64] utilisent la vision stéréoscopique pour calculer la direction de pointage du doigt, en 3D. Ils comparent deux modes de pointage, l'orientation du doigt et la droite de vue entre l'œil et le bout du doigt. Leurs expérimentations montrent que la deuxième solution est préférable. WU *et al.* [136] proposent une méthode avec une seule caméra. La position 3D du doigt est calculée à partir des positions de l'épaule et du coude, en supposant que l'utilisateur fait face à la caméra.

1. *Hidden Markov Models*

Ainsi, deux catégories de systèmes se distinguent dans la littérature : ceux basés sur le suivi d'un ou plusieurs doigts sur une surface 2D, et ceux utilisant le doigt pour pointer en 3D.

Nous proposons, dans ce chapitre, d'expérimenter le suivi multi-doigts en 3D, avec deux caméras. L'objectif est de trouver une méthode rapide et efficace pour calculer les trajectoires 3D des doigts. Nous privilégions une approche par apparence, dans une optique temps réel ; les approches par modèle 3D étant d'une complexité trop importante pour les applications visées. Le filtre de KALMAN est utilisé pour estimer les positions 3D avec des observations bruitées. Les erreurs sur les observations proviennent de différentes sources.

6.1.1 *Suivi de mouvement*

D'une manière générale, le suivi du mouvement d'un objet, ou d'une cible, est un processus nécessitant plusieurs étapes :

- La *mesure*, qui consiste à mesurer une propriété dans l'image pour caractériser l'objet.
- La *validation*, qui détermine la validité de la mesure en se basant sur des connaissances a priori, ou sur la prédiction.
- L'*estimation*, qui met à jour la position de l'objet.
- La *prédiction*, qui calcule la position de l'objet dans l'image suivante.

La position de l'objet peut être exprimée dans le repère de la scène, ou dans le repère des images. Cette position peut être une coordonnée en 2D dans l'image, les coordonnées de la boîte englobante de l'objet, les statistiques du second ordre de la distribution spatiale des pixels, ou un ensemble de points correspondant au contour de la cible.

WU *et al.* [138] proposent de découpler l'estimation globale de la posture de la main, de l'estimation des articulations des doigts. Un apprentissage avec un gant numérique (`CyberGlove`, cf. paragraphe 3.1.1.2) leur permet de modéliser les contraintes entre les articulations. Ils modélisent aussi les déformations de la main à partir d'un ensemble de 28 configurations de base.

L'algorithme *Mean shift* (COMANICIU *et al.* [22]) est une procédure itérative, utilisée pour la détermination de la position la plus probable de la cible dans la prochaine image. Une mesure de similarité entre le modèle et les candidats est utilisée. BRADSKI [11] a proposé une variante, appelée *Camshift* (*Continuously Adaptive Mean-Shift*), qui permet de trouver le centre et la taille d'un objet en se basant sur sa distribution de couleur. Les probabilités sont calculées avec un histogramme de la couleur de la peau.

Les contours actifs, ou « *snakes* » , permettent de suivre un objet avec un contour déformable. Le contour est déformé en minimisant différents termes d'énergies (interne, externe). Cette méthode a été utilisée par HEAP ET SAMARIA [56], avec un « modèle de distribution de points »[2], mais elle ne permet que de suivre une main ouverte.

Le filtre de KALMAN (WELCH ET BISHOP [133]) est fréquemment utilisé pour le suivi en vision par ordinateur. Il permet de mettre en oeuvre les étapes de validation et de prédiction, et également de lisser les mesures au cours du temps. Il peut être utilisé avec différents modèles de mouvement (position, vitesse,

2. *Point Distribution Model* (PDM)

accélération) pour filtrer la position de la main, ou des doigts (Oka *et al.* [100]). Shamaie et Sutherland [115] utilisent le filtrage sur les coordonnées de la boîte englobante, pour gérer les occultations et croisements lors de mouvements avec les deux mains. Nous verrons, au paragraphe 6.2.2, l'utilisation du filtre de Kalman pour le suivi tridimensionnel des doigts.

Le filtre particulaire, ou algorithme de « *Condensation* » (*Conditional Density Propagation*), de Isard et Blake [67], est aussi très utilisé pour le suivi (Black et Jepson [8]). Par exemple, Bretzner *et al.* [15] utilisent des caractéristiques de couleur et un modèle hiérarchique de la main.

6.1.2 *Sources d'erreur*

Dans un cas idéal, la vision stéréoscopique doit pouvoir donner la position 3D exacte de chaque point, à partir de ses projections dans les deux vues. Toutefois, les positions 3D ainsi calculées manquent de précision. Plusieurs sources d'erreur perturbent le calcul de la position 3D :

- la détection des doigts est parfois peu précise, à cause par exemple d'une mauvaise segmentation ;
- la discrétisation des images implique que la localisation d'un point se fait à un pixel près, ce qui peut représenter plusieurs millimètres lors de la triangulation 3D ;
- les deux caméras ne sont pas synchronisées, il existe donc un décalage temporel entre l'acquisition des deux images. Pendant cet intervalle de temps, le doigt se déplace proportionnellement à la vitesse de son mouvement. Par conséquent, la triangulation est biaisée, principalement au niveau de la profondeur (axe optique des caméras).

Le filtre de Kalman permet de réduire l'influence de ces erreurs, en lissant les trajectoires 3D estimées.

6.2 SUIVI TRIDIMENSIONNEL DES DOIGTS

La connaissance du nombre et des positions des doigts est suffisante pour certaines applications, notamment celles de type « souris 3D » avec le geste de pointage du doigt. Le nombre de doigts permet de distinguer différents modes d'interaction (pointage ou commande), et peut être associé à des actions telles que les clics d'une souris.

À partir de la position des doigts dans chacune des deux vues, il est possible de calculer leurs positions 3D par triangulation. L'estimation des positions 3D manque de précision pour diverses raisons (paragraphe 6.1.2). Pour estimer les positions 3D avec des observations bruitées, nous utilisons un filtre de Kalman pour chaque doigt. Ainsi, les trajectoires sont lissées, et l'erreur d'estimation est réduite. De plus, la détection des doigts est améliorée par une recherche locale basée sur la prédiction du filtre de Kalman.

Le suivi des bouts des doigts est divisé en deux parties :

- Étude du *geste de pointage* (ou *geste déictique**) : un seul doigt est utilisé pour pointer. Le but étant de localiser la position de ce doigt, il n'est pas nécessaire de traiter l'image entière. Nous proposons de restreindre la détection du bout du doigt à une zone de recherche, grâce au suivi et à la

**cf.* § 1.1 *p.* 2

prédiction de la position à partir de la paire d'images précédente, ce qui permet de diminuer le temps de calcul.
- Étude du *suivi multi-doigts* : deux étapes d'appariement de points se rajoutent au système de suivi d'un doigt :
  - *appariement stéréoscopique* : les détections dans chaque vue sont mises en correspondance, afin de calculer les positions 3D des doigts ;
  - *appariement temporel* : les détections 3D calculées à l'étape précédente sont associées aux filtres de KALMAN correspondant, afin d'estimer les trajectoires 3D.

6.2.1 *Filtre de* KALMAN

Le filtre de KALMAN [73] est la technique la plus fréquemment utilisée pour suivre le mouvement d'un objet. Une introduction détaillée a été réalisée par WELCH ET BISHOP [133]. Le filtre de KALMAN est une solution optimale au problème du filtrage linéaire de données. Il estime l'état d'un objet à partir d'observations bruitées, en minimisant l'*erreur quadratique moyenne*.

Le filtre permet d'estimer la position de l'objet en réalisant un compromis entre la position observée dans l'image, et la position prédite à partir du modèle de mouvement. Ainsi, le filtre permet d'analyser les positions passées, d'estimer la position actuelle avec des observations bruitées, et de prédire la position future de l'objet. L'algorithme est récursif, avec deux grandes étapes : une étape de *prédiction*, à partir de la dernière estimation de l'état, et une étape de *correction*, utilisant une mesure pour corriger l'état prédit.

6.2.1.1 Processus à estimer

Le filtre de KALMAN estime l'état $\mathbf{x} \in \mathbb{R}^n$ d'un processus discret, modélisé par l'équation linéaire suivante :

$$\mathbf{x}_{k+1} = \mathrm{A}\mathbf{x}_k + \mathbf{w}_k \tag{6.1}$$

avec A la matrice de transition de l'état k à l'état $k+1$ modélisant le mouvement, et $\mathbf{w}_k$ une variable aléatoire représentant le bruit du processus.

L'observation, ou mesure, est décrite par le vecteur $\mathbf{z} \in \mathbb{R}^m$. L'équation de mesure est la suivante :

$$\mathbf{z}_k = \mathrm{H}\mathbf{x}_k + \mathbf{v}_k \tag{6.2}$$

avec H la matrice d'observation, reliant l'état $\mathbf{x}_k$ à la mesure $\mathbf{z}_k$, et $\mathbf{v}_k$ une variable aléatoire représentant le bruit de mesure.

Les bruits du processus et de mesure sont indépendants, de distributions normales et blanches, avec respectivement Q la covariance du bruit du processus, et R la covariance du bruit de mesure :

$$\mathbf{w}_k = \mathcal{N}(0, \mathrm{Q}) \tag{6.3}$$
$$\mathbf{v}_k = \mathcal{N}(0, \mathrm{R}) \tag{6.4}$$

6.2.1.2 Équations de mise à jour

Notons $\hat{\mathbf{x}}_k$ l'*estimation de l'état*, on cherche à minimiser l'erreur $\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$, dont la covariance est $\mathrm{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T]$. L'*estimation de l'état a priori* $\hat{\mathbf{x}}_{k+1}^-$, ou *prédiction*, peut être calculée par la formule suivante :

$$\hat{\mathbf{x}}_{k+1}^- = \mathrm{A}\hat{\mathbf{x}}_k \tag{6.5}$$

L'*erreur d'estimation a priori* est donnée par $\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-$, et a pour covariance $\mathrm{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}]$. La solution donnée par le filtre de KALMAN pour l'estimation de l'état est :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - \mathrm{H}\hat{\mathbf{x}}_k^-) \tag{6.6}$$

où K_k est le gain de KALMAN, et $\mathbf{z}_k - \mathrm{H}\hat{\mathbf{x}}_k^-$ est le terme d'*innovation*. L'innovation correspond à la différence entre la mesure $\mathbf{z}_k$ et la mesure prédite $\mathrm{H}\hat{\mathbf{x}}_k^-$. Pour trouver une solution optimale à cette équation, le gain K_k est choisi pour minimiser la covariance de l'erreur P_k.

Finalement, les équations pour les étapes de prédiction et de correction sont les suivantes :

– Équations de prédiction :

$$\hat{\mathbf{x}}_{k+1}^- = \mathrm{A}\hat{\mathbf{x}}_k \tag{6.7}$$

$$\mathrm{P}_{k+1}^- = \mathrm{A}\mathrm{P}_k\mathrm{A}^T + \mathrm{Q} \tag{6.8}$$

– Équations de correction :

$$K_k = \mathrm{P}_k^-\mathrm{H}^T(\mathrm{H}\mathrm{P}_k^-\mathrm{H}^T + \mathrm{R})^{-1} \tag{6.9}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - \mathrm{H}\hat{\mathbf{x}}_k^-) \tag{6.10}$$

$$\mathrm{P}_k = (\mathrm{I} - K_k\mathrm{H})\mathrm{P}_k^- \tag{6.11}$$

6.2.1.3 Modèle de mouvement uniforme à accélération constante

La matrice A modélise le mouvement de l'objet. Le modèle de mouvement utilisé est généralement à vitesse constante ou à accélération constante. Le mouvement de la main n'étant pas uniforme, ce type de modèle n'est pas adapté pour décrire tous les mouvements de manière générale. Toutefois, nous supposons que les gestes que nous considérons ici ont une dynamique adaptée.

Dans la suite de ce chapitre, nous utilisons le modèle de mouvement uniforme à accélération constante, pour le suivi d'un point en trois dimensions. Le vecteur d'état $\mathbf{x}_k$ s'écrit :

$$\mathbf{x}_k = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z})^T \tag{6.12}$$

avec (x, y, z) la position, $(\dot{x}, \dot{y}, \dot{z})$ la vitesse, et $(\ddot{x}, \ddot{y}, \ddot{z})$ l'accélération du point.

La matrice A s'écrit :

$$\mathrm{A} = \begin{pmatrix} 1 & 0 & 0 & dt & 0 & 0 & \frac{dt^2}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 & \frac{dt^2}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & \frac{dt^2}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.13}$$

L'observation que nous fournissons au filtre est la position mesurée du point, la matrice H s'écrit :

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.14}$$

6.2.1.4 Choix des matrices de covariances

Une étude réalisé par WUEST [139] sur différents modèles de mouvements illustre l'influence des différentes matrices de covariances. Pour le choix du bruit du processus, il faut trouver un compromis entre l'inertie de l'estimation et le bruit résultant : si Q est grand, l'adaptation à de nouveaux états est rapide, mais l'estimation est bruitée ; et inversement si Q est petit, l'estimation est plus précise, mais l'adaptation aux changements est plus lente. La valeur initiale de P doit être élevée, afin que l'estimation converge rapidement.

Nous supposons que les trois composantes (X, Y, Z) de l'espace 3D sont indépendantes, ce qui permet d'avoir des matrices de covariance diagonales. Nous supposons dans notre modèle que l'accélération est constante, ce qui n'est pas forcément vrai. Par conséquent, nous supposons que la covariance du bruit du processus est importante sur les composantes de vitesse et d'accélération, tandis qu'elle est faible sur les composantes de position.

Les variances du bruit de mesure R sont calculées avec une séquence d'images où le doigt reste fixe. Le résultat montre que l'erreur de mesure est plus importante sur la composante de profondeur Z, que sur X et Y :

$$Var(X, Y, Z) = (2.31, 2.39, 15.06)$$

Cette différence entre les trois composantes s'explique par le fait que Z correspond à l'axe optique des caméras, donc à la dimension de profondeur du système stéréoscopique. Lors de la triangulation, l'estimation sur cette composante est moins précise que pour les deux autres.

6.2.2 *Suivi d'un doigt*

Nous nous intéressons tout d'abord au suivi tridimensionnel de la position d'un doigt pour le geste de pointage, avec un filtre de KALMAN. Puisque nous cherchons juste à localiser la position du doigt, il n'est pas nécessaire de traiter l'image entière. Nous proposons donc de réduire la détection du bout du doigt à une zone de recherche [3], grâce au suivi du doigt et à la prédiction de sa position à partir de la paire d'images précédente. Cette méthode a plusieurs intérêts :

- améliorer la robustesse du système en facilitant la localisation du doigt (si le doigt est bien détecté au début de la séquence),
- lisser les trajectoires obtenues en estimant les positions 3D parmi les observations bruitées,
- combler un manque d'observation en utilisant la prédiction,
- réduire fortement le temps de calcul.

3. *Region Of Interest* (ROI)

6.2.2.1 Algorithme développé

Nous utilisons le modèle de mouvement uniforme à accélération constante, présenté dans la section précédente. La figure 6.1 résume les différentes étapes du traitement :

1. Le doigt est détecté avec la méthode décrite dans la paragraphe 4.3.3, dans la paire d'images stéréoscopiques, ce qui fournit les mesures 2D.
2. Les mesures 2D dans chaque vue permettent de calculer la position 3D du bout du doigt, utilisée comme mesure 3D pour le filtre de KALMAN.
3. À partir du calcul de la mesure 3D avec une paire d'images, le filtre de KALMAN permet d'estimer l'état, et de prédire la position 3D correspondant à la paire d'images suivante.
4. La position 3D prédite est projetée dans chacune des deux images, avec les matrices de projection des deux caméras, pour obtenir une prédiction 2D de la position du bout du doigt.
5. Les prédictions 2D permettent de réduire la zone de recherche du bout du doigt dans les images. La taille utilisée pour la fenêtre de recherche est, par exemple, de 80×80 pixels.
6. La détection du bout du doigt est alors réalisée dans cette fenêtre de recherche. S'il n'est pas détecté, la recherche est étendue à toute l'image.
7. Enfin, la *contrainte épipolaire** est vérifiée pour s'assurer que les détections du bout du doigt dans les deux images correspondent. Si cette contrainte n'est pas validée, la détection est relancée sur toute l'image.

*cf. § B.3 p. 128

6.2.2.2 Résultats

Nous avons présenté notre configuration matérielle au section 2.2. La source d'erreur la plus importante est due au fait que les caméras ne soient pas synchronisées. Par conséquent, le doigt peut bouger entre l'acquisition des deux images (paragraphe 6.1.2), et la triangulation est biaisée, principalement au niveau de la profondeur (dans la direction correspondant à l'axe optique des caméras).

Afin de pouvoir mesurer les erreurs sur le calcul de la position 3D, il est nécessaire de connaître la *vérité terrain*, c'est-à-dire de connaître à chaque instant la position 3D du bout du doigt dans le repère de la scène. Or, ceci n'est pas possible à moins d'utiliser un autre système permettant de mesurer cette position 3D.

Dans notre configuration, l'erreur de reconstruction se retrouve principalement sur la composante Z, correspondant à la profondeur (axes optiques). Par conséquent, afin de pouvoir mesurer les erreurs de reconstruction 3D dans la dimension de la profondeur, nous avons réalisé des tests avec des trajectoires planes. Ces trajectoires sont réalisées dans un plan perpendiculaire aux axes optiques, en l'occurrence un cercle et un carré tracés sur le bureau, correspondant au plan $z = 0$ (figure 6.2).

La figure 6.2 montre les détections du doigt sur une séquence, cumulées sur une image, qui fournissent les mesures 2D. La figure 6.3 montre les résultats pour le cercle et le carré, avec les trajectoires 3D estimées en rouge, et les mesures en bleu. On constate que l'erreur de reconstruction se retrouve principalement

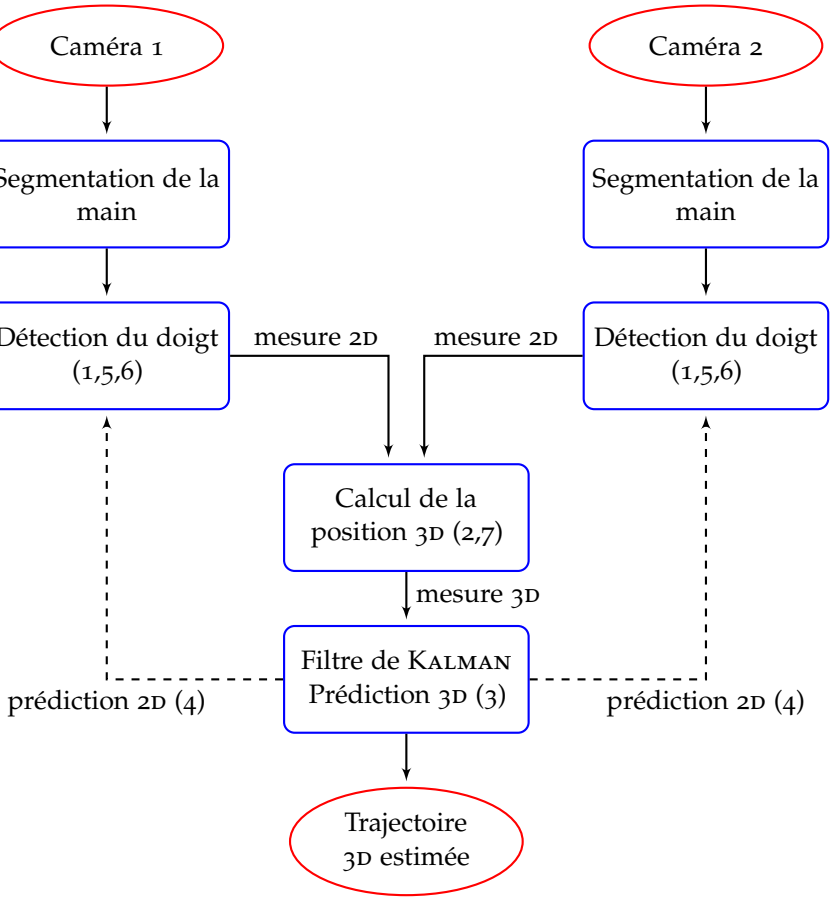


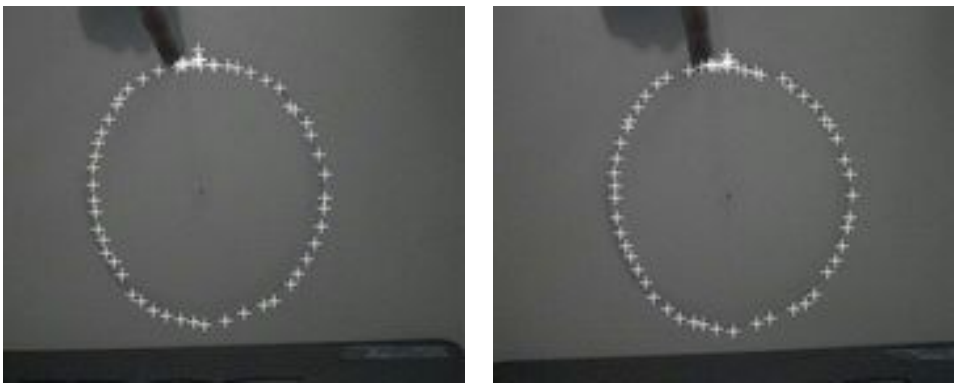
FIGURE 6.1 – *Schéma récapitulatif des différentes étapes pour le suivi 3D d'un doigt, avec un filtre de* KALMAN. *Les numéros correspondent aux étapes du suivi présentées au paragraphe 6.2.2.1.*

sur la composante Z, correspondant à la profondeur. Le filtre de KALMAN lisse la trajectoire 3D, et diminue l'écart-type sur la profondeur, qui passe de $9,77$ à $5,46$ dans le cas du cercle. De plus, grâce à la réduction de la zone de recherche, les temps de calcul sont largement améliorés et dépassent le temps réel (> 30 Hz). Cependant, l'erreur sur la composante Z dépend aussi de la vitesse du mouvement.

INFLUENCE DE LA VITESSE DU MOUVEMENT

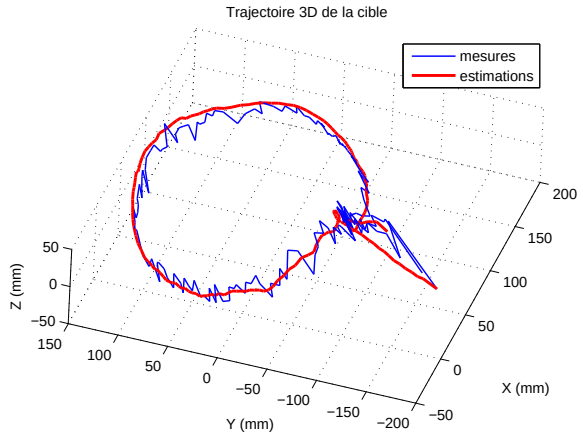
La vitesse du mouvement influence l'erreur de reconstruction, du fait de l'absence de synchronisation. En effet, plus le mouvement est rapide, plus le doigt est susceptible de se déplacer entre l'acquisition de deux images successives, d'où une erreur plus importante lors de la triangulation.

Le tableau 6.1 illustre ceci avec l'étude de deux trajectoires planes (cercle et carré), traitées en temps réel (30 Hz). Celles-ci sont réalisées avec trois vitesses différentes. Ainsi, une trajectoire plus rapide est composée d'un nombre de points plus faible. L'écart-type sur la composante Z est ensuite calculé pour comparer les erreurs de reconstruction. Dans les deux cas l'écart-type augmente avec la vitesse, et il est plus faible pour la trajectoire estimée par le filtre de KALMAN que pour les mesures. On observe aussi que l'écart-type de l'estimation

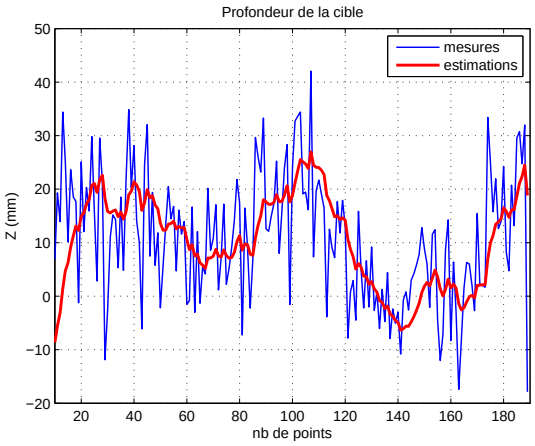


(a) *image gauche* (b) *image droite*

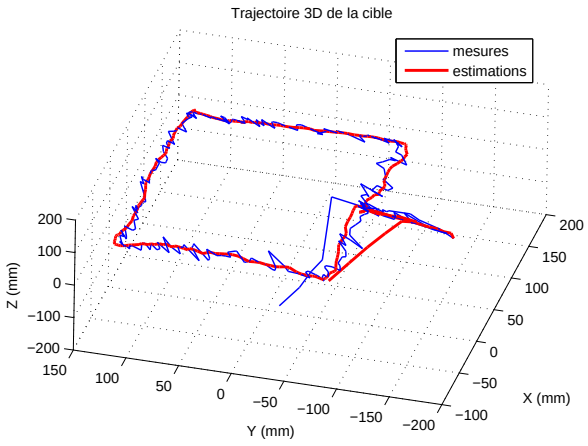
FIGURE 6.2 – *Images gauche et droite avec l'ensemble des positions détectées sur une séquence.*



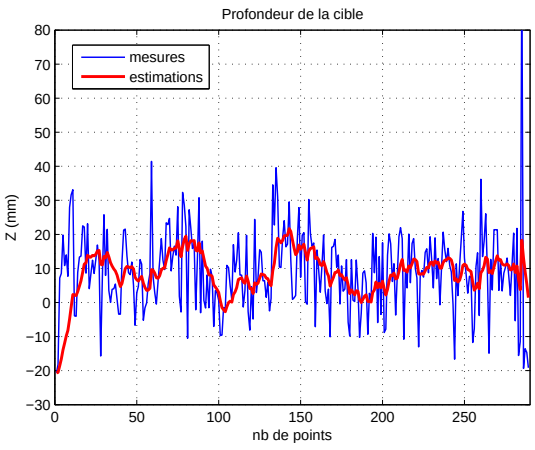
(a) *cercle, 3D*



(b) *cercle, composante Z*



(c) *carre, 3D*



(d) *carre, composante Z*

FIGURE 6.3 – *Reconstruction 3D pour deux trajectoires, un cercle et un carré, réalisées dans le plan du bureau ($Z = 0$). Les coordonnées sont en millimètres, les trajectoires estimées en rouge, et les mesures en bleu.*

TRAJECTOIRE	VITESSE	NOMBRE DE POINTS	ÉCART-TYPE MESURES	ÉCART-TYPE ESTIMATION
Cercle	lent	306	9,77	5,46
	moyen	189	11,32	8,39
	rapide	108	14,76	10,83
Carré	lent	290	10,48	4,87
	moyen	185	11,15	4,43
	rapide	106	12,28	6,04

TABLEAU 6.1 – *Évolution de l'écart-type sur la profondeur (composante Z) en fonction de la vitesse de réalisation du mouvement : plus le geste est réalisé rapidement, plus le nombre de points de la trajectoire est faible, et plus l'écart-type est important.*

est plus faible pour le carré, ce qui s'explique par le fait que le modèle de mouvement est plus adapté à une trajectoire linéaire.

6.2.2.3 Synthèse

Nous avons présenté un système de suivi tridimensionnel d'un doigt de la main en temps réel, qui permet une détection plus robuste en la réduisant à une zone de recherche, et une réduction de l'erreur d'estimation en lissant les trajectoires 3D. Cette méthode offre une solution à l'estimation de la trajectoire 3D pour le geste de pointage, adaptée à un système pour lequel le temps de calcul est primordial. Les positions obtenues peuvent être utilisées pour la reconnaissance de trajectoires.

Notre algorithme résout les occultations sur une courte période : si le doigt n'est pas détecté dans une image, nous utilisons sa prédiction donnée par le filtre de KALMAN. Toutefois, ceci ne fonctionne que si le mouvement du doigt est linéaire, ou approximativement linéaire sur une courte durée.

6.2.3 *Suivi multi-doigts*

Nous nous intéressons maintenant à la généralisation de la méthode présentée dans la section précédente, pour le suivi des doigts de la main. Il est donc possible d'avoir jusqu'à cinq détections dans chaque image, voire plus si l'on prend en compte la possibilité de fausses détections. Le problème est alors de mettre en correspondance les détections des doigts dans les deux vues pour calculer la position 3D de chaque doigt. Il faut ensuite associer chaque mesure 3D au filtre de KALMAN correspondant, afin de calculer la trajectoire de chaque doigt.

L'étape d'appariement entre les deux vues est fondamentale en vision stéréoscopique : un point physique 3D se projette en un point dans l'image gauche, et un point dans l'image droite. Pour résoudre le problème inverse, et reconstruire la position 3D des points, il faut déterminer les couples de points gauche/droite correspondant à un même point 3D. Une fois qu'un couple de points est apparié, la reconstruction 3D consiste en une triangulation des droites de vues (sec-

tion B.3). Pour faire cette mise en correspondance, nous utilisons des contraintes géométriques : les *droites épipolaires* et la *contrainte d'ordre*.

Une fois que les détections des deux vues sont appariées, les mesures 3D sont calculées. Pour suivre chaque doigt avec un filtre de KALMAN, il est aussi nécessaire d'apparier chaque mesure 3D avec le filtre du doigt correspondant. Pour réaliser cet appariement, nous utilisons une minimisation des distances entre les mesures et les prédictions.

6.2.3.1 Appariement stéréoscopique

Avec les détections des doigts dans les deux vues, nous pouvons calculer leurs positions 3D dans le repère de la scène. Mais pour cela, il faut mettre en correspondance les détections entre les deux vues, afin de savoir quelles détections correspondent au même point physique. C'est un problème classique de vision stéréoscopique, connu sous le nom d'*appariement stéréoscopique* ou de *mise en correspondance*.

Pour résoudre ce problème, nous utilisons les contraintes données par la vision stéréoscopique :

CONTRAINTE D'UNICITÉ : chaque point d'une image admet au plus un correspondant dans l'autre image.

CONTRAINTE D'ORDRE : l'ordre des points est conservé dans les images.

CONTRAINTE ÉPIPOLAIRE : le correspondant d'un point d'une image est sur une droite dans l'autre image (cf. paragraphe B.3.3).

Notons :

$\{p_i\}, 1 \leq i \leq N$	les bouts des doigts détectés dans l'image gauche,
$\{a_i\}, 1 \leq i \leq N$	les droites épipolaires associées aux doigts détectés dans l'image gauche,
$\{q_j\}, 1 \leq j \leq M$	les bouts des doigts détectés dans l'image droite.
$\{b_j\}, 1 \leq i \leq M$	les droites épipolaires associées aux doigts détectés dans l'image droite,

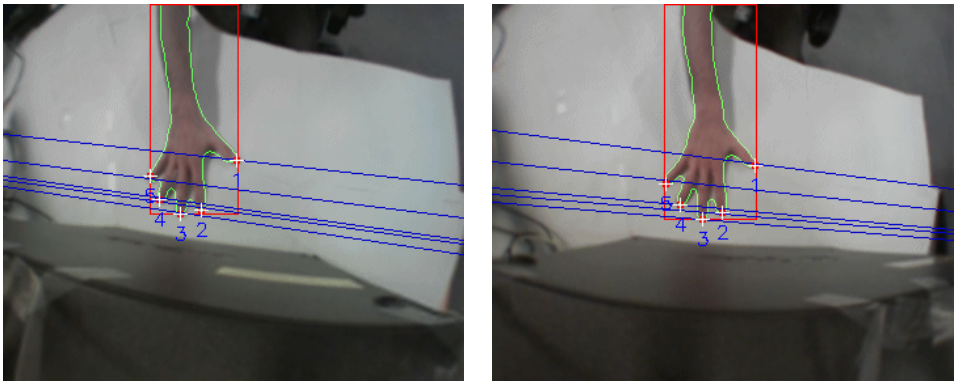
Nous calculons une matrice D des distances entre les doigts détectés dans une image et les droites épipolaires des doigts détectés dans l'autre image :

$$\mathrm{D}(i,j) = d(p_i, b_j) + d(q_j, a_i) \tag{6.15}$$

où d est la distance entre un point et une droite.

Cette matrice permet de réduire les candidats possibles pour l'appariement, mais elle ne permet qu'un appariement point par point, sans tenir compte des relations entre les doigts. C'est pourquoi nous utilisons la contrainte d'ordre, pour apparier les points en tenant compte des appariements voisins, et donc obtenir une mise en correspondance globale. La contrainte d'ordre est associée à l'ordre des points sur le contour. Lors de cette étape, si des points n'ont pas de correspondant dans l'autre vue, ils sont considérés comme des fausses détections et sont alors rejetés.

La figure 6.4 montre un exemple de mise en correspondance, dans lequel les cinq doigts ont été correctement appariés, y compris lorsque les droites épipolaires sont très proches.



(a) *image gauche* (b) *image droite*

FIGURE 6.4 – *Mise en correspondance des doigts entre les deux images, avec les détections et les droites épipolaires (les numéros correspondent aux appariements).*

6.2.3.2 Suivi tridimensionnel

Chaque bout de doigt est suivi en trois dimensions avec un filtre de KALMAN, avec le modèle de mouvement présenté au paragraphe 6.2.1.3. Les positions 3D des bouts des doigts, calculées par l'étape d'appariement stéréoscopique, forment l'ensemble de mesures pour les filtres de KALMAN. Un appariement est nécessaire pour associer les détections 3D au filtre de KALMAN du doigt correspondant. Par ailleurs, il peut arriver qu'un doigt soit mal détecté ou occulté par un autre doigt. Dans ce cas, nous utilisons la prédiction du filtre de KALMAN correspondant. Si un nouveau doigt est détecté et qu'il ne correspond à aucun des doigts déjà suivis, un nouveau filtre de KALMAN est créé.

Étant donné que nous effectuons le suivi de plusieurs doigts simultanément, il nous faut associer chaque mesure 3D avec le filtre de KALMAN du doigt correspondant. Pour faire la mise en correspondance, nous utilisons une minimisation de la distance euclidienne 3D entre les mesures et les prédictions.

Les mesures $\mathbf{z}_k^i$, $i \in [1, M]$, sont les positions 3D détectées dans la paire d'images à l'instant k. Le filtre de KALMAN de chaque doigt suivi $\mathbf{x}_k^j$, $j \in [1, N]$, est mis à jour avec la détection $\mathbf{z}_k^i$ qui minimise la distance euclidienne 3D entre les prédictions et les détections :

$$I = \arg\min_i d(\hat{\mathbf{x}}_k^{-j}, \mathbf{z}_k^i) \tag{6.16}$$

Toutefois, cette méthode ne permet pas de prendre en compte les relations entre les doigts. Nous rajoutons donc une contrainte sur l'ordre des doigts, de la même façon que pour l'appariement stéréoscopique.

Si le nombre de mesures est différent du nombre de doigts suivis, il est intéressant de considérer l'appariement d'un point de vue plus global. La solution que nous avons retenue consiste à tester les différentes combinaisons possibles, en calculant la somme des distances pour chaque combinaison, et en prenant en compte l'ordre des doigts. La meilleure séquence d'appariement est celle qui minimise la distance globale.

Enfin, s'il y a plus de mesures que de doigts suivis, un filtre de KALMAN est initialisé pour les mesures qui n'ont pas été appariées. Si un doigt n'est pas détecté dans une paire d'images, nous utilisons la prédiction du filtre

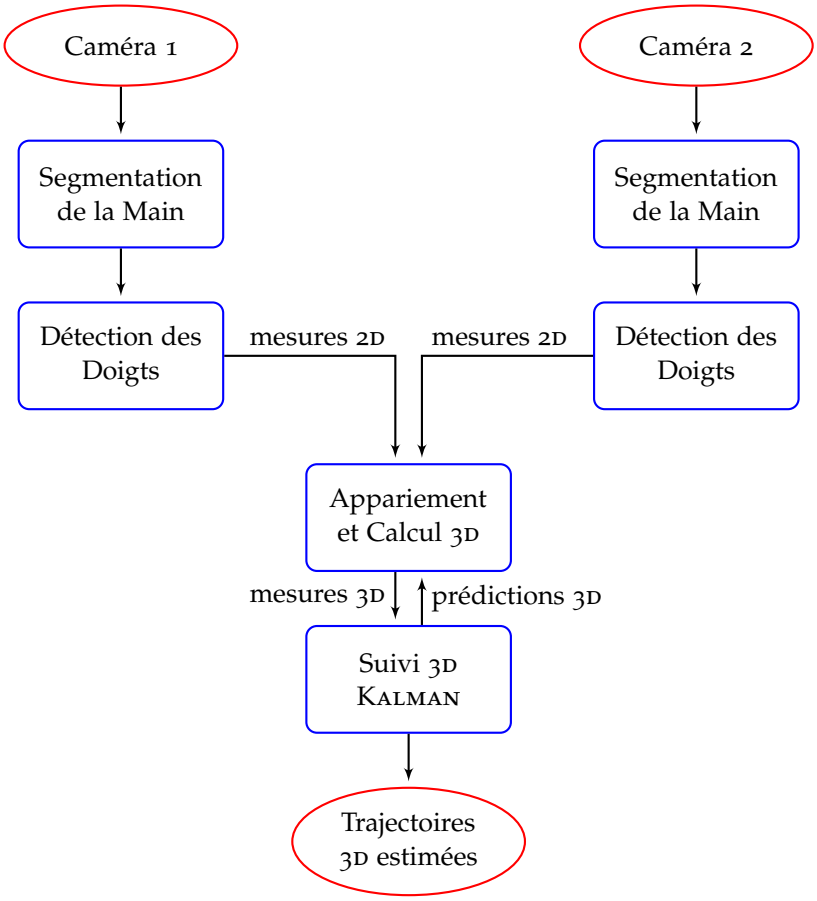


FIGURE 6.5 – *Schéma récapitulant les différentes étapes pour le suivi 3D de plusieurs doigts, avec un ensemble de filtres de* KALMAN.

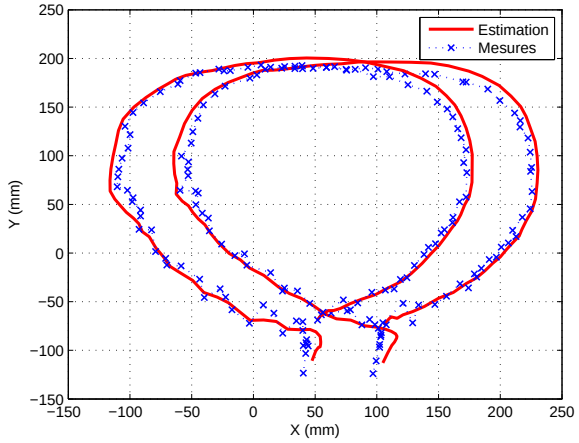
de KALMAN correspondant pendant quelques images, au cas où le doigt soit simplement occulté et réapparaisse dans les images suivantes. Sinon, il est supprimé.

6.2.3.3 Résultats

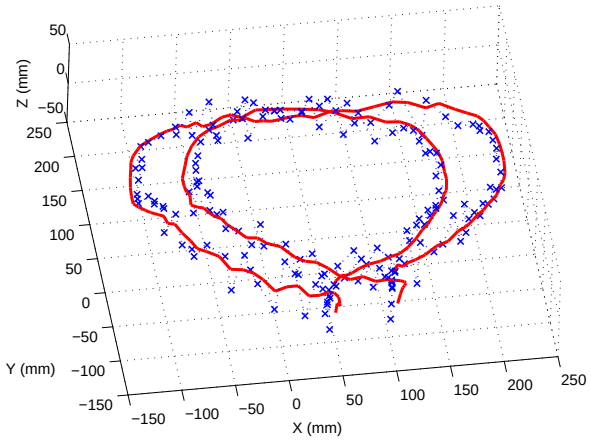
De la même façon que pour le suivi d'un seul doigt, l'absence de synchronisation entre les caméras introduit des erreurs dans le calcul des positions 3D. Ce problème est plus pénalisant dans le cas du suivi multi-doigt, puisque les doigts sont très proches. Il est donc nécessaire d'avoir une estimation précise de leur position pour les différencier et les suivre séparément.

La figure 6.6 montre les trajectoires obtenues pour des gestes circulaires, réalisés avec deux et trois doigts, dans le plan (X, Y). Ces figures montrent que les différentes étapes se sont bien déroulées, bien que parfois un des doigts n'ai pas été détecté, et que les croisements de trajectoires sont bien gérés.

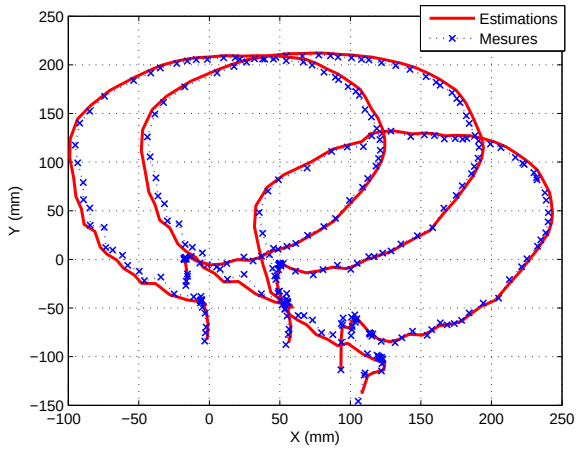
Cependant ce n'est pas toujours le cas, notamment lorsque le mouvement est trop rapide (figure 6.6e). La limitation du système vient du fait que chaque doigt est considéré indépendamment des autres. Par conséquent si le mouvement de la main est trop rapide, il peut y avoir des conflits dans la mise à jour des filtres de KALMAN. Lorsqu'il y a quatre ou cinq doigts, il est difficile de bien apparier les détections avec les filtres.



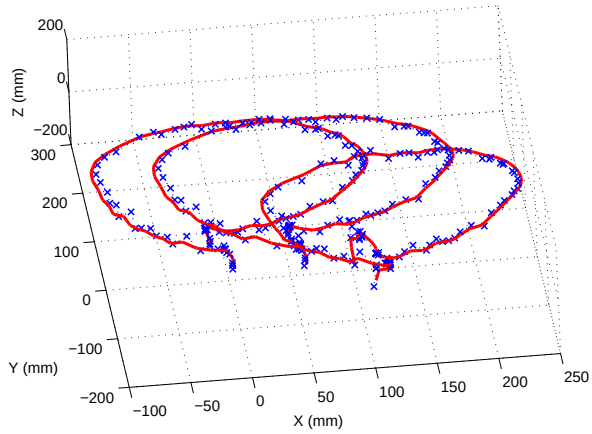
(a) *avec 2 doigts*



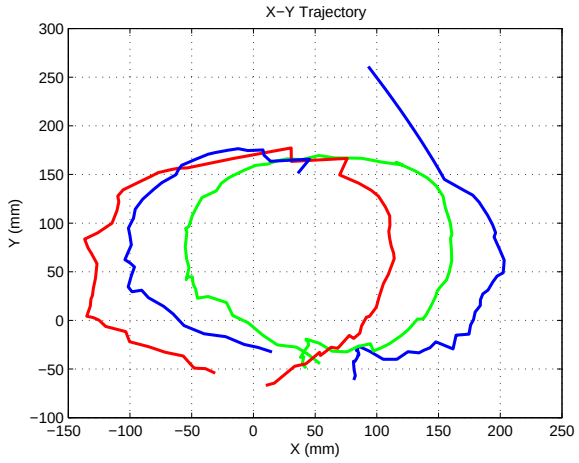
(b) *avec 2 doigts en 3D*



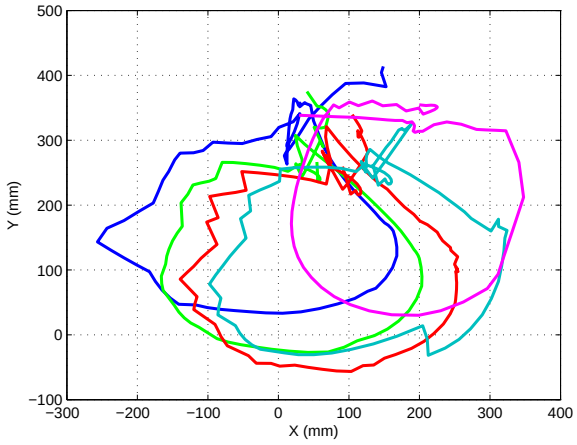
(c) *avec 3 doigts*



(d) *avec 3 doigts en 3D*



(e) *avec 3 doigts*



(f) *avec 5 doigts*

FIGURE 6.6 – *Reconstruction 3D pour des gestes circulaires avec 2 ou 3 doigts, réalisés dans le plan du bureau ($Z = 0$) et représentés en 2D et en 3D. Les coordonnées sont en millimètres, les trajectoires estimées en rouge, et les mesures en bleu.*

6.3 SUIVI 2D AVEC UN MODÈLE SQUELETTIQUE

Pour effectuer la mise en correspondance entre les doigts détectés dans les deux vues, il est utile d'avoir des informations sur la géométrie de la main, ce qui est possible en utilisant les contraintes morphologiques naturelles qui existent entre les doigts. Pour cela, nous utilisons un modèle squelettique de la main, afin de prendre en compte la géométrie de la main et les relations entre les doigts.

Notre méthode de suivi est basée sur les étapes suivantes :

A. *Segmentation* de la main avec l'histogramme de la couleur de peau, avec l'algorithme présenté au paragraphe 4.2.6.

B. *Extraction de points et de valeurs caractéristiques* : centre, poignet et bouts des doigts, présentés à la section 4.3, auxquels nous rajoutons la « base du doigt » et la détection du pouce.

C. *Recalage* du modèle géométrique du squelette de la main, en utilisant les caractéristiques.

6.3.1 *Modèle squelettique*

Notre méthode de suivi consiste à utiliser un modèle squelettique simplifié de la main, qui est recalé grâce à des points caractéristiques extraits dans l'image courante. Ces points correspondent aux articulations et aux extrémités du squelette simplifié de la main :
- l'extrémité du bras E,
- le poignet P,
- le centre de la main C,
- les bouts de chaque doigt D_i,
- la base de chaque doigt B_i, correspondant à l'articulation principale du doigt avec la main (articulation MCP [4]),
- le pouce.

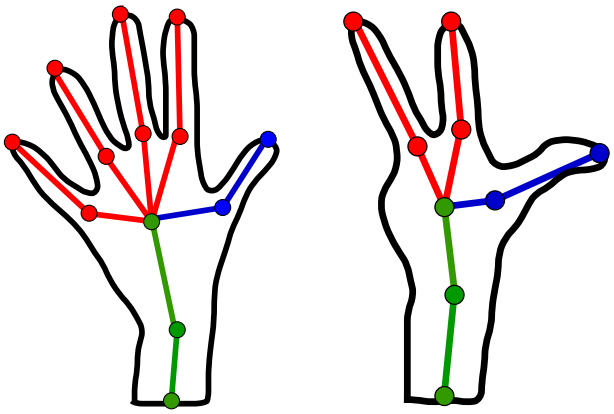
Nous pouvons représenter le squelette (figure 6.7) avec le vecteur correspondant à l'avant-bras, $\mathbf{CP}$ (en vert), la liaison entre le poignet et le centre de la main, $\mathbf{PE}$ (en vert), et les vecteurs de chaque doigt $\mathbf{CB_i}$ et $\mathbf{B_iD_i}$ (en rouge) ainsi que du pouce (en bleu).

6.3.2 *Caractéristiques*

Afin de recaler le modèle squelettique dans les images, il est nécessaire de connaître la position de certains points caractéristiques de la main. Nous avons présenté le calcul de certains de ces points à la section 4.3 : le centre de la main et les bouts des doigts. Nous avons aussi vu qu'il était possible de détecter les creux entre chaque doigt. Nous allons utiliser la position de ces creux pour calculer la position d'un point situé à la « base du doigt ».

Nous proposons également une méthode pour déterminer si l'un des doigts détectés est le pouce. Connaître la position du pouce peut servir à distinguer une main gauche d'une main droite. Savoir si le pouce est présent est aussi

4. Articulation métacarpo-phalangienne



(a) (b)

FIGURE 6.7 – *Modèle squelettique de la main*

une information intéressante pour la reconnaissance de gestes, puisqu'il peut permettre de distinguer différents modes d'interaction.

6.3.2.1 Base du doigt

Nous utilisons les détections correspondant aux creux entre chaque doigt pour calculer un point à la « base du doigt », correspondant à l'articulation MCP [5], entre la paume et la première phalange. Notons :

$$\begin{cases} D_i, & 1 \leq i \leq N, & \text{les bouts des doigts ;} \\ C_i, & 1 \leq i \leq N-1, & \text{les creux ;} \\ B_i, & 1 \leq i \leq N, & \text{le point « base du doigt ».} \end{cases}$$

Pour chaque doigt D_i, on détermine le creux le plus proche en terme de distance sur le contour, soit C_i soit C_{i+1}. On calcule le symétrique de ce point par rapport à l'axe du doigt (par exemple le point C_2' sur la figure 6.8). Le point « base du doigt », B_i, est le milieu en ces deux points.

6.3.2.2 Détection du pouce

Une fois les doigts détectés, un test est effectué pour déterminer si l'un des doigts est le pouce. Ce test est basé sur le calcul de l'angle entre le vecteur **CP** et la base de chaque doigt (vecteur $\mathbf{CB_i}$), le point C étant le centre de la main, et le point P correspondant au poignet.

Comme le montre la figure 6.7 cet angle est proche de 90° pour le pouce. Le pouce est détecté si cet angle est compris entre 80° et 100°. Si le pouce a déjà été détecté dans l'image précédente, on se base alors sur la méthode de suivi afin de continuer à connaître la position du pouce même quand il ne fait pas un angle proche de 90°.

5. Articulation métacarpo-phalangienne

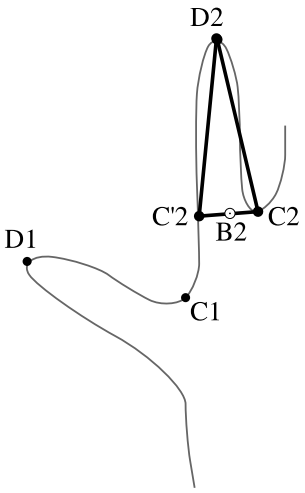


FIGURE 6.8 – *Méthode pour le calcul de la position du point B_2 (« base du doigt ») à partir des creux détectés (C_i)*

6.3.3 *Recalage du modèle*

Le suivi de la main dans le flux vidéo est réalisé en mettant à jour les paramètres du modèle avec chaque nouvelle image. Pour cela, nous utilisons les caractéristiques extraites des images.

6.3.3.1 Initialisation

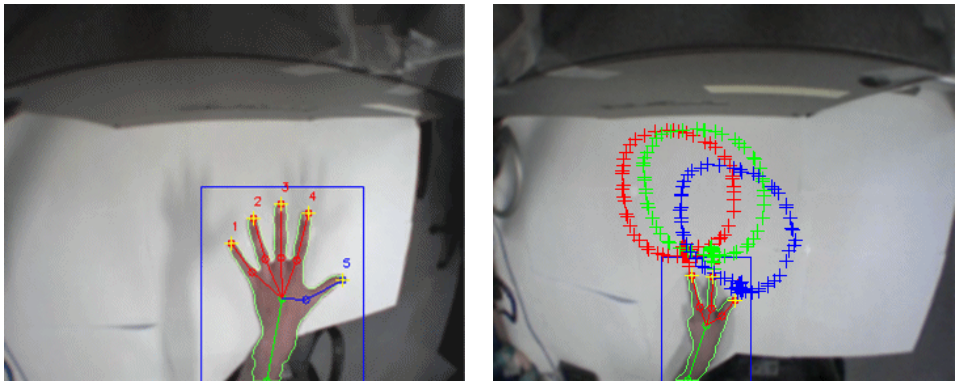
Un problème classique des méthodes de suivi par modèle est l'initialisation de ce modèle. Cette étape est généralement effectuée avec la main à plat, doigts écartés, ce qui permet d'ajuster le modèle à la morphologie de la main de l'utilisateur. Un des avantages de notre méthode est de ne pas avoir besoin de faire d'hypothèses sur la configuration de la main lors de son entrée dans le champ de vision de la caméra. Lorsqu'un ou plusieurs doigts sont détectés, le modèle est initialisé.

6.3.3.2 Recalage global

Dans chaque nouvelle image, le modèle calculé à l'image précédente est recalé en se basant sur les détections dans la nouvelle image : d'abord la direction de l'avant-bras, puis le vecteur **CP** donnant l'orientation de la main. Ceci permet de suivre le mouvement global du bras et fournit une estimation de la position des doigts.

6.3.3.3 Recalage des doigts

La mise à jour de la position des doigts est alors effectuée en recherchant pour chaque doigt du modèle la détection correspondante. Cet appariement de points se fait par une minimisation globale de la distance euclidienne entre les points, en testant les différentes combinaisons possibles. Chaque point ne peut être apparié qu'une seule fois.



(a) (b)

FIGURE 6.9 – *Résultat du suivi avec le modèle squelettique : (a) le modèle squelettique recalé sur la main, et la fenêtre de recherche (en bleu), et (b) trajectoires des doigts obtenues.*

Lorsqu'une détection ne correspond à aucun doigt, un nouveau doigt est créé. Lorsqu'un doigt n'est pas mis à jour, c'est-à-dire qu'aucune détection ne lui est appariée, sa position est gardée en mémoire pendant quelques images au cas où sa disparition ne soit que temporaire (occultation, erreur de détection ou de segmentation). Si après quelques images (paramètre à régler) il n'a toujours pas été mis à jour, alors il est supprimé.

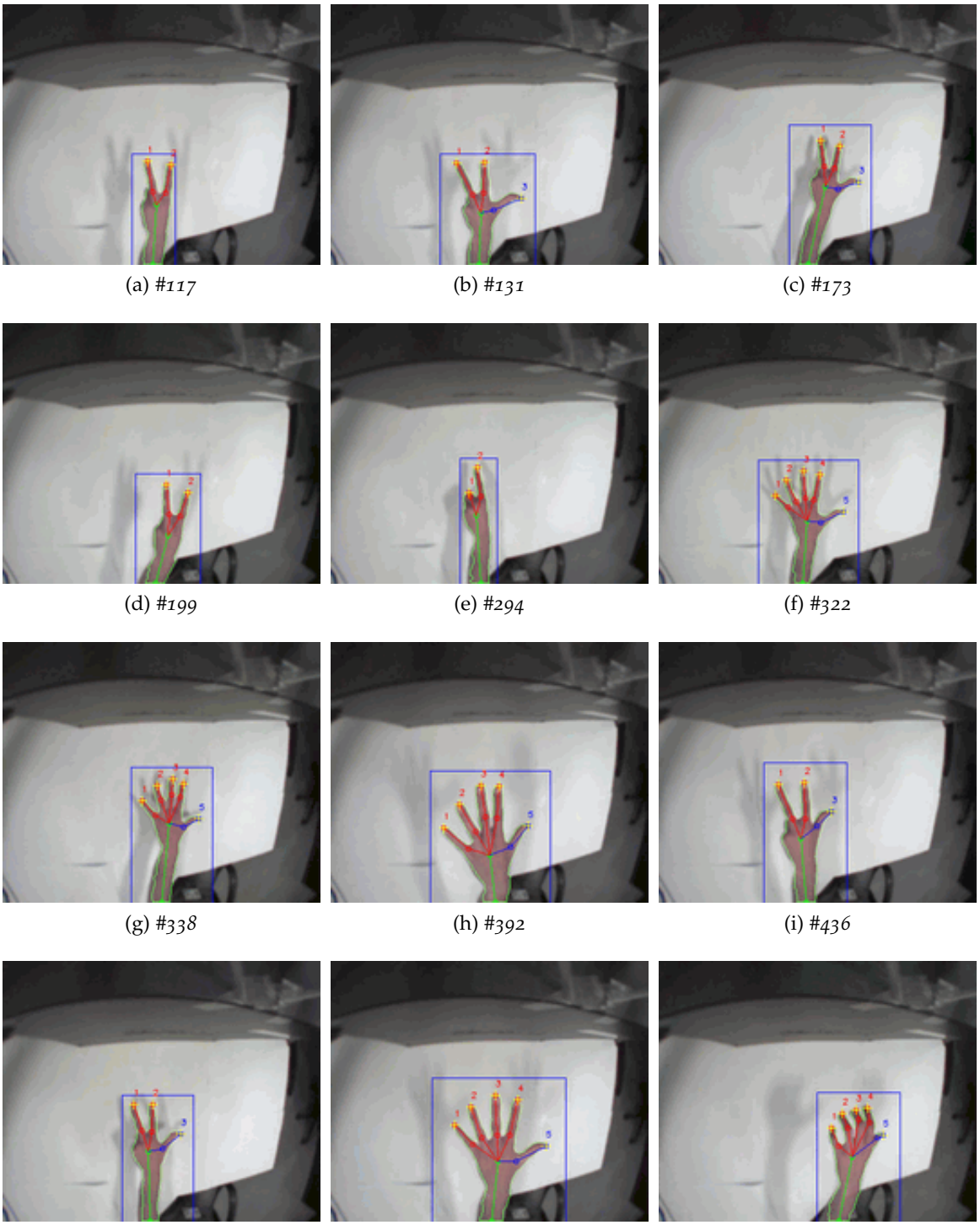
6.3.4 *Résultats et discussion*

La figure 6.9a montre le résultat du recalage du modèle sur une image. Les cinq doigts sont bien détectés, le pouce l'est aussi. Le rectangle bleu représente la fenêtre de recherche utilisée pour faciliter la segmentation de l'image et réduire les temps de calculs. La figure 6.9b montre les trajectoires de trois doigts, obtenues par le suivi. Les détections sont associées au bon doigt, ce qui permet de reconstituer leur trajectoire au fil du temps.

La figure 6.10 montre le résultat du suivi par recalage, sur une séquence d'images dans laquelle le nombre de doigts varie. Le modèle s'adapte bien au nombre de doigts présents, et le pouce est bien détecté. Le temps de traitement est en moyenne de 25 images/sec, il peut varier suivant la taille de la fenêtre de recherche.

Notre méthode permet un suivi en temps réel du mouvement de la main, grâce au fait que le modèle est très simplifié. Le modèle permet aussi d'associer chaque détection au doigt correspondant, et ainsi de reconstituer les trajectoires des doigts au cours du temps. Notre modèle à l'avantage d'être rapide et efficace, et d'être peu contraignant pour l'utilisateur :

- le centre de la main est bien détecté, même si l'utilisateur a le bras nu,
- la détection ne dépend pas de la position et de l'orientation (suivant l'axe Z) de la main dans l'image,
- aucune supposition n'est faite sur la scène (zone d'entrée de la main par exemple),
- aucune hypothèse sur la main (gauche ou droite),
- une fenêtre de recherche limite le temps de calcul,
- le temps de calcul est très faible.



(a) #117 (b) #131 (c) #173

(d) #199 (e) #294 (f) #322

(g) #338 (h) #392 (i) #436

(j) #458 (k) #517 (l) #531

FIGURE 6.10 – *Suivi de la main avec le modèle squelettique* 2D *sur une séquence vidéo. Chaque doigt est correctement identifié alors que la posture évolue au fil de la séquence.*

Toutefois, la simplicité de notre modèle implique des limitations sur le suivi de certains gestes, par exemple si des doigts sont « collés » ou qu'ils se touchent, auxquels cas on ne détecte qu'un seul doigt ; de même si certains doigts sont en partie pliés. Pour résoudre ce problème, nous avons envisagé d'utiliser des informations sur les dimensions des doigts, largeur et longueur. La largeur peut être calculée au bout des doigts, pour être robuste aux erreurs de segmentation, en prenant la distance entre les points $P(i-k)$ et $P(i+k)$, représentés sur la figure 4.14a. La longueur est calculée en utilisant le point « base du doigt ».

Ainsi, en connaissant la largeur des doigts, il est possible de détecter si deux doigts sont collés ; et la connaissance de la longueur des doigts permet de savoir s'ils sont en partie repliés. En effet, en limitant les degrés de liberté des articulations et en considérant les contraintes existantes entre celles-ci, nous pouvons négliger l'articulation IPD [6] entre la phalangine et la phalangette. Ainsi, on peut résumer à trois les configurations principales de la main : soit le doigt est *tendu*, soit il est *plié* à moitié, soit il est complètement *replié*. Les autres configurations intermédiaires sont assimilées à la position « doigt plié ».

Cependant, le problème de l'invariance aux changements d'échelle se pose. Il est donc nécessaire de normaliser les dimensions calculées par une distance telle que celle entre le poignet et le centre. De plus, il est nécessaire d'effectuer un apprentissage pour adapter le modèle aux dimensions de la main de l'utilisateur, ce qui nécessite une initialisation « main ouverte ». Une autre possibilité est d'utiliser des dimensions moyennes telles que celles proposées par DELAMARRE ET FAUGERAS [37].

Les différentes expérimentations menées dans ce sens n'ont pas été concluantes, les dimensions calculées n'étant pas suffisamment fiables. Nous nous sommes ensuite intéressés à l'apport d'une deuxième vue afin de reconstruire le squelette en 3D et d'utiliser des contraintes morphologiques supplémentaires.

6.4 SUIVI 3D

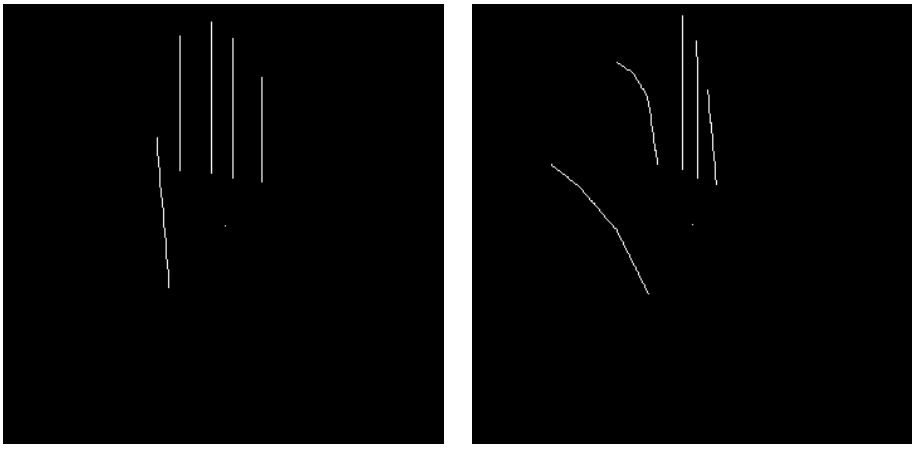
Le modèle présenté dans la section précédente permet de suivre la main dans un flux vidéo en ayant une bonne connaissance de la configuration des doigts lorsque ceux-ci sont bien séparés. En appliquant ce suivi sur deux vues, il est possible de calculer le modèle 3D correspondant. Toutefois, le problème de la visualisation des résultats se pose. Nous avons donc élaboré un modèle 3D volumique afin de visualiser la configuration de la main obtenue. De plus, ce modèle 3D permet d'ajouter des informations sur les dimensions de la main, et sur les contraintes entre les articulations.

Notre approche consiste donc à calculer le modèle 3D à partir des squelettes extraits dans chaque vue, et à utiliser les contraintes du modèle 3D pour éviter les configurations aberrantes.

6.4.1 *Calcul du squelette 3D*

À partir de la section précédente, il est possible de suivre la main dans chacune des deux vues en calculant le modèle squelettique 2D. Si le nombre de doigts détectés est le même dans les deux vues, il est alors facile d'obtenir un

6. Articulation inter-phalangienne distale



(a) (b)

FIGURE 6.11 – *Squelette 3D de la main après fusion des informations issues des deux vues (représenté ici dans une image après projection).*

modèle squelettique 3D en calculant les points caractéristiques 3D du squelette (poignet, centre de la main, bases et bouts des doigts) par triangulation à partir des positions de ces points dans les deux vues.

La difficulté est de déterminer la configuration du modèle 3D si le nombre de doigts détectés n'est pas le même dans les deux vues. Dans ce cas, nous pouvons utiliser la configuration du modèle 3D à l'instant précédent, afin de déterminer celle des deux vues qui donne le bon nombre de doigts. Deux cas de figure se présentent alors :

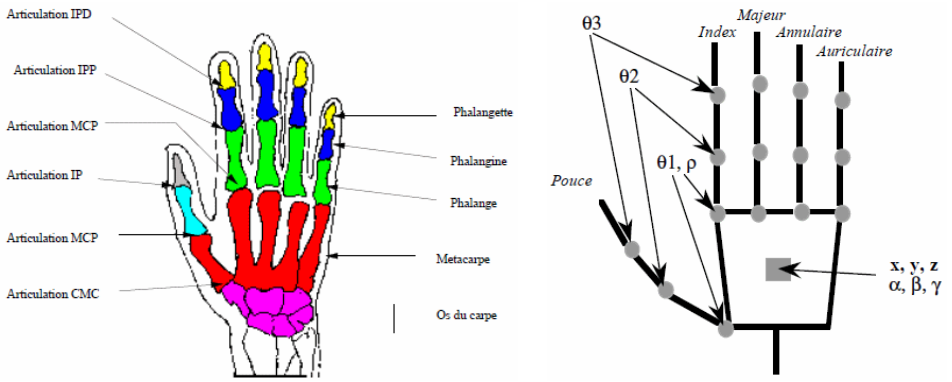
- s'il manque un doigt dans une des deux vues, sa position 3D est interpolée grâce au modèle 3D ;
- s'il y a une détection de trop, nous testons les différents appariements possibles et nous gardons celui qui est le plus proche du modèle 3D à l'instant précédent.

Nous obtenons ainsi un squelette 3D de la main (figure 6.11). Pour visualiser le résultat du suivi en 3D, nous avons développé un modèle 3D volumique. De plus, ce modèle permet de prendre en compte des contraintes morphologiques de la main, telles que les relations entre les articulations des doigts.

6.4.2 *Modèle 3D articulé de la main*

La main humaine est un système biomécanique complexe. Le squelette de la main (figure 6.12a) est composé des os du poignet, les métacarpes, et des os des doigts, les phalanges, phalangines et phalangettes. Les articulations entre ces os portent les noms suivants :

- l'articulation de chaque doigt sur son métacarpe est appelée *métacarpo-phalangienne* (MCP),
- les articulations entre phalanges sont appelées *inter-phalangiennes proximales* (IPP) et *inter-phalangiennes distales* (IPD),
- pour le pouce, les articulations entre phalanges sont appelées *inter-phalangiennes* (IP),
- l'articulation située à la base du pouce est dite *carpo-métacarpienne* (CMC).



(a) (b)

FIGURE 6.12 – *Anatomie de la main : (a) squelette de la main avec le détail des différentes articulations (extrait de* OUHADDI ET HORAIN [104]*), et (b) les degrés de liberté de la main (source :* BRAFFORT [12]*).*

6.4.3 *Notre modèle 3D volumique*

Nous nous sommes inspiré des nombreux travaux sur le suivi par modèle 3D (section 3.5) pour élaborer notre modèle. Pour ce faire, il faut déterminer les dimensions, les degrés de liberté et les relations entre les articulations.

6.4.3.1 Dimensions

Pour éviter un apprentissage visant à adapter le modèle à la main de l'utilisateur, nous utilisons des dimensions basées sur les proportions physiques du modèle de main développé par DELAMARRE ET FAUGERAS [37] (figure 6.13).

6.4.3.2 Degrés de libertés

La main est généralement modélisée par 27 degrés de liberté (REHG ET KANADE [111], cf. section 3.5). Étant donné les gestes que nous considérons, nous avons défini un modèle 3D légèrement simplifié : les degrés de libertés des articulations IPD ne sont pas pris en compte et l'articulation CMC du pouce n'est décrite que par 2 degrés de libertés. Nous avons donc retenu 21 degrés de libertés :

- 3 translations et 3 rotations au niveau de la paume (et donc de la main entière) ;
- 1 flexion et 1 abduction au niveau de l'articulation CMC du pouce (assimilée aux articulations MCP des autres doigts) ;
- 1 flexion et 1 abduction au niveau des articulations MCP des quatres autres doigts (cf. figure 6.14) ;
- 1 flexion pour chacune des 5 articulations IPP. les articulations IPD sont considérées comme liées aux articulations IPP[7], leurs degrés de libertés sont déterminés par les contraintes présentées dans la section suivante.

6.4.3.3 Contraintes

Les différentes contraintes physiologiques limitant les mouvements de la main ont été étudiées par LEE ET KUNII [86], DELAMARRE ET FAUGERAS [37] et

7. les articulations IP du pouce sont assimilées aux articulations IPP et IPD des autres doigts.

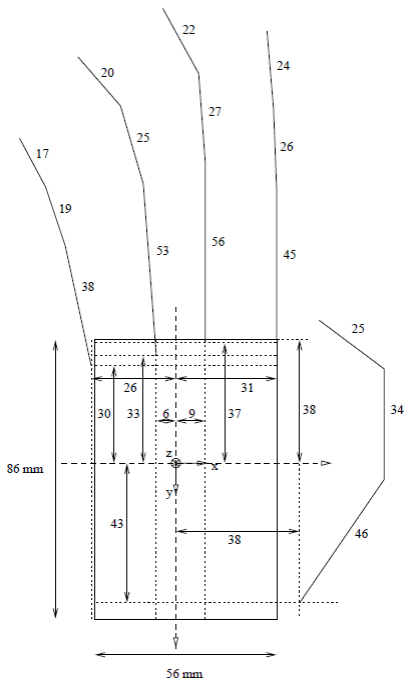


FIGURE 6.13 – *Proportions du modèle squelettique développé par* DELAMARRE ET FAUGERAS [37].

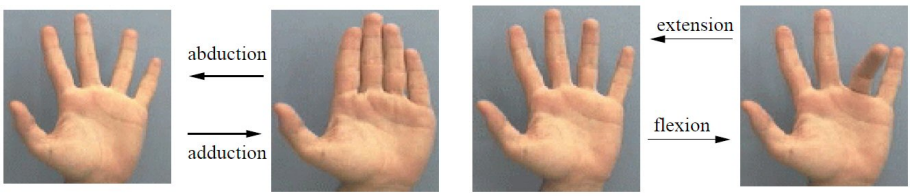


FIGURE 6.14 – *Mouvements réalisés par les doigts : (a) abduction et adduction, et (b) flexion et extension (extrait de* OUHADDI ET HORAIN [104]*).*

OUHADDI ET HORAIN [104]. Les mouvements des doigts de la main sont régis par des contraintes biomécaniques qui font que certaines postures ne sont pas réalisables. Ces contraintes sont de deux types : statiques ou dynamiques. À cause de sa morphologie particulière, les mouvements du pouce sont soumis à un ensemble de contraintes différent des autres doigts.

CONTRAINTES STATIQUES

Les contraintes statiques traduisent les limites des angles d'abduction et adduction ou de flexion et extension des différentes articulations :

- angles maximums de flexion des articulations du pouce :

$$\begin{cases} \text{Phalange :} & -60° < \theta < 25° \\ \text{Phalangine :} & -40° < \theta < 0° \\ \text{Phalangette :} & -70° < \theta < 25° \\ \text{Abduction CMC :} & 5° < \theta < 40° \end{cases} \tag{6.17}$$

(a) (b) (c)

FIGURE 6.15 – *Notre modèle 3D volumique permettant de visualiser la configuration de la main.*

- angles maximums de flexion des autres doigts, pour les *phalanges* :

$$\begin{cases} \text{Index :} & -90° < \theta < 10° \\ \text{Majeur :} & -95° < \theta < 10° \\ \text{Annulaire :} & -100° < \theta < 10° \\ \text{Auriculaire :} & -105° < \theta < 10° \end{cases} \tag{6.18}$$

- angles maximums de flexion des différents doigts, pour les *phalangines* :

$$-95° < \theta < 0° \tag{6.19}$$

CONTRAINTES DYNAMIQUES

Les contraintes dynamiques représentent les relations entre les articulations des doigts. Ces relations sont définies entre les articulations d'un même doigt et entre les articulations des phalanges de doigts voisins. Pour simplifier, nous considérons uniquement les flexions des articulations d'un même doigt.

La relation liant les flexions des articulations IP et MCP du pouce est :

$$\theta_{IP} = 2\,\theta_{MPC} \tag{6.20}$$

L'abduction au niveau de l'articulation MCP du pouce, proposée dans certains travaux, est négligée. Pour les autres doigts, la relation entre les flexions de la phalangette et de la phalangine est :

$$\theta_{IPD} = \frac{2}{3}\,\theta_{IPP} \tag{6.21}$$

6.4.4 *Résultats*

Le modèle 3D a été élaboré avec la bibliothèque OpenGL [8]. Il est basé sur l'assemblage d'éléments géométriques simples : un pavé pour la paume de la main et des cylindres pour les diverses parties des doigts, des articulations et des contraintes ont également été définies. Ce modèle permet de visualiser en 3D la configuration de la main (figure 6.15). La figure 6.16 montre le résultat du suivi 3D de la main avec le modèle volumique. On constate que le suivi est bien réalisé lorsque les doigts sont bien séparés.

8. SILICON GRAPHICS INC. : OpenGL the industry's foundation for high performance graphics. http://www.opengl.org

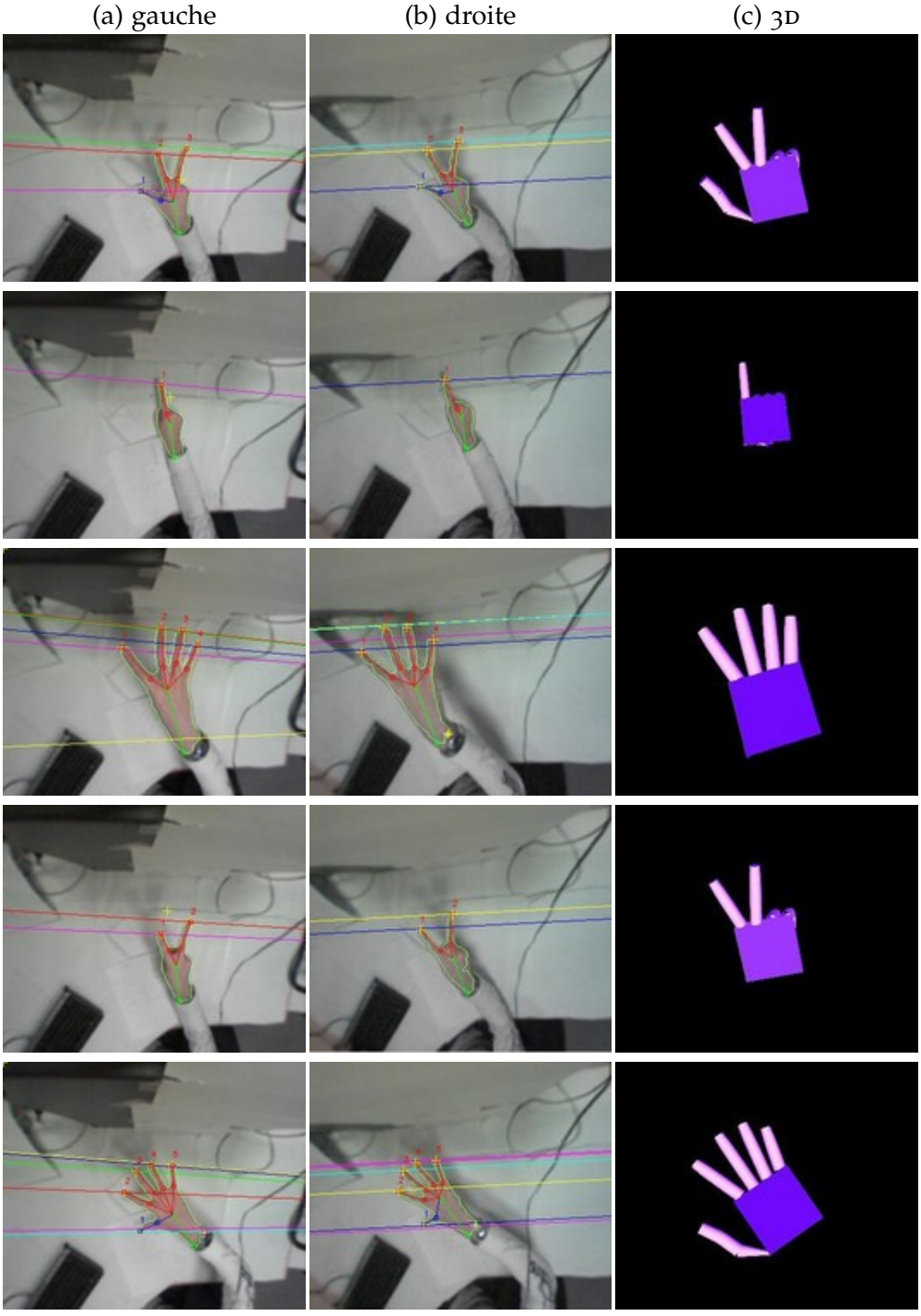


FIGURE 6.16 – *Séquences d'images résultats du suivi*

6.5 RÉSUMÉ

Nous avons présenté des méthodes pour le suivi tridimensionnel des doigts et de la main [24, 25, 26]. Nous nous sommes d'abord intéressés au suivi des doigts basé sur le filtrage de KALMAN 3D, ce qui permet de réduire l'erreur d'estimation en lissant les trajectoires 3D. En effet, nous avons vu que l'absence de synchronisation entre les deux caméras provoque une erreur importante sur la reconstruction 3D, et plus particulièrement sur la composante de profondeur. De plus, il peut arriver qu'un des doigts ne soit pas détecté ou soit occulté. Le filtre de KALMAN permet de combler ce manque d'observation. Cette méthode a d'abord été mise au point pour le geste de pointage. Le doigt est détecté dans une zone de recherche centrée sur les projections de la position 3D prédite, ce qui permet de réduire le temps de calcul et de faciliter la détection.

Dans un deuxième temps, nous avons étendu cette approche au suivi multi-doigts, avec un filtre de KALMAN pour chaque doigt. Une étape d'appariement stéréoscopique est réalisée pour mettre en correspondance les détections dans les deux vues. Ensuite, les mesures 3D obtenues sont affectées aux filtres de KALMAN pour le suivi des doigts. Les résultats montrent que cette approche fonctionne avec plusieurs doigts, mais la mise en correspondance est délicate si le mouvement est trop rapide. Par ailleurs, le filtre de KALMAN n'est pas parfaitement adapté à des mouvements circulaires ou d'autres mouvements plus complexes, du fait du modèle de mouvement linéaire. Il serait donc intéressant de modéliser le mouvement non-linéaire par un filtre de KALMAN étendu.

Pour améliorer le suivi, nous avons ajouté des informations sur la géométrie de la main, avec un modèle squelettique. Le modèle est recalé avec des points caractéristiques détecté dans l'image : le centre de la main, le poignet, les bouts et bases des doigts. Cette méthode permet un suivi en temps réel du mouvement de la main, grâce au fait que le modèle est très simplifié. Le modèle permet aussi d'associer chaque détection au doigt correspondant, et ainsi de reconstituer les trajectoires des doigts au cours du temps. Toutefois, ce modèle ne permet pas de savoir si des doigts sont pliés ou dépliés ou si deux doigts sont collés.

Le suivi par modèle squelettique a été étendu en 3D, en combinant les informations issues des deux vues. Un modèle 3D volumique a été développé afin de visualiser le résultat. De plus, ce modèle fournit des informations supplémentaires sur la morphologie de la main et notamment sur les relations entre les articulations des doigts. On obtient ainsi la posture 3D de la main et les positions des bouts des doigts lorsque ceux-ci sont séparés.

7 CONCLUSION

Dans cette thèse, nous nous sommes intéressés à la réalisation d'un système de suivi tridimensionnel de la main et de reconnaissance de gestes, basé sur la vision stéréoscopique, dans le cadre de la conception d'une interface Homme-Machine 3D. Nous avons étudié les différentes composantes d'un tel système et nous avons proposé des solutions tenant compte de contraintes applicatives importantes, avec notamment le traitement temps réel du flux vidéo. Les problèmes traités concernent la détection de la main dans un flux vidéo, l'extraction de caractéristiques représentant la forme et la position de la main, la reconnaissance de postures parmi un vocabulaire et le suivi tridimensionnel du mouvement des doigts et de la main, avec deux caméras. Cette synthèse retrace les principales conclusions de cette étude et les contributions de notre travail pour aboutir à un démonstrateur. Nous donnons ensuite quelques pistes pour la poursuite de nos travaux.

Nous avons tout d'abord présenté le domaine de l'interprétation visuelle des gestes de la main dans le cadre de l'interaction homme-machine, et nous avons évoqué les contraintes liées à la vision par ordinateur en général, et au contexte industriel de cette thèse en particulier. L'objectif de cette thèse était d'étendre le concept du système `3DFeel`, afin de permettre une interaction gestuelle en trois dimensions plus naturelle et intuitive, et la moins contraignante possible pour les utilisateurs (pas de périphérique supplémentaire, uniquement la main et les caméras). De plus, nous avons pris en considération des contraintes matérielles, sur le type et la disposition des caméras, et sur l'objectif de traitements temps réel, robustes aux conditions d'acquisition et d'illumination.

Nous avons ensuite proposé un ensemble de méthodes permettant d'atteindre ces objectifs. La première étape concerne la détection de la main dans un flux vidéo avec une méthode robuste aux ombres. En effet, celles-ci provoquent des erreurs de segmentation avec les méthodes par différence d'images, y compris en environnement intérieur. La méthode retenue consiste à modéliser la couleur de la peau par un histogramme adaptatif dans l'espace C_bC_r, mis à jour périodiquement afin de s'adapter aux variations de luminosité. De plus, l'apprentissage est automatisé grâce à l'utilisation de seuils C_bC_r sur les premières images du flux vidéo, ce qui permet d'adapter le modèle à l'utilisateur. Une réduction de l'espace de recherche permet d'accélérer les traitements.

La deuxième partie de notre travail concerne la reconnaissance de postures 2D de la main. Nous avons étudié et comparé plusieurs descripteurs de formes, afin de calculer un vecteur de caractéristiques représentant la forme de la main, en prenant en compte les invariances aux transformations euclidiennes (translation, rotation et changement d'échelle). Pour évaluer et comparer les résultats de reconnaissance, nous avons procédé à l'acquisition d'une base de données constituée de 11 gestes réalisés par 18 personnes. Cette base est représentative des gestes pouvant être utilisés dans notre application, et de la variabilité de la forme de la main et des gestes dans l'espace de travail en fonction des utilisateurs. Nous avons ensuite proposé des améliorations en prenant

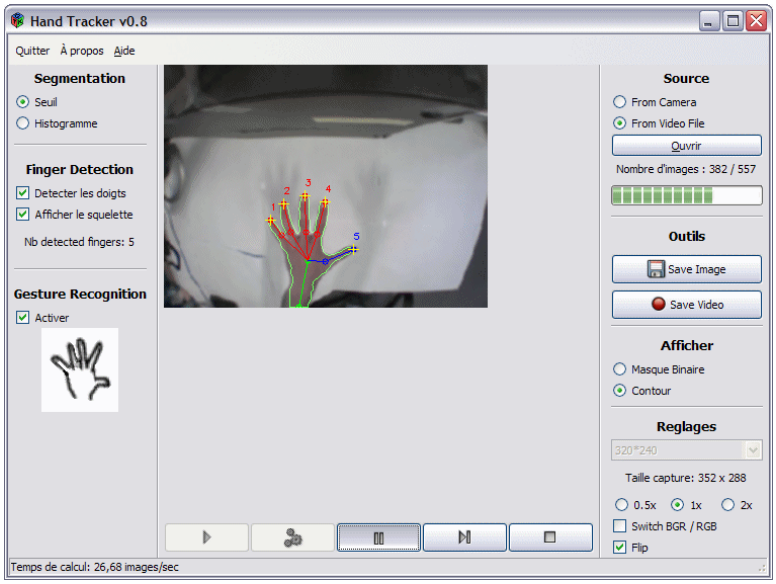


FIGURE 7.1 – *Interface graphique réalisée pour évaluer le suivi et la reconnaissance de gestes.*

en compte l'information temporelle, afin de rejeter les gestes pour lesquels la reconnaissance est incertaine, et en utilisant l'information supplémentaire fournie par une deuxième vue de la scène.

Enfin, nous avons considéré le problème du suivi tridimensionnel des doigts et de la main, basé sur la vision stéréoscopique avec deux caméras calibrées. Dans un premier temps, nous avons abordé le suivi 3D des doigts dans le cadre de la réalisation de gestes de pointage pour remplacer la souris. Notre méthode est basée sur le filtrage de KALMAN pour estimer la position 3D avec des observations bruitées et utiliser la prédiction 3D pour rendre le suivi plus robuste. L'absence de synchronisation entre les deux caméras provoque des erreurs sur le calcul de la position 3D, en particulier sur la composante de profondeur. Cette approche a été étendue au suivi multi-doigts, en effectuant un appariement stéréoscopique entre les doigts détectés dans chaque vue et en associant les mesures 3D au filtre de KALMAN correspondant. Toutefois, cette méthode est rapidement mise en défaut lorsque le mouvement des doigts est trop rapide, à cause notamment du manque d'informations sur les relations entre les doigts.

Le manque d'information sur la morphologie de la main nous a conduit à modéliser son squelette de façon simplifiée. Notre modèle squelettique prend en compte les articulations principales. Il est recalé dans les images grâce à la détection de points caractéristiques : le centre de la main, le poignet, les bouts et les « bases » des doigts. Cette approche a d'abord été menée en 2D, puis a été étendue en 3D en combinant le suivi dans les deux vues. Le squelette 3D ainsi calculé a été utilisé pour synthétiser un modèle 3D volumique. Ce dernier prend en compte des contraintes morphologiques supplémentaires sur les relations entre les articulations des doigts, ce qui permet d'éviter des configurations irréalisables.

Une interface graphique intégrant ces différentes méthodes a été réalisée (figure 7.1) et utilisée pour des démonstrations. Ainsi, lors de la « semaine des applications » à STMICROELECTRONICS, des dizaines de personnes ont pu tester

le système, avec la segmentation et le suivi de la main, et la reconnaissance de postures, tous ces traitements étant réalisés en temps réel sur un PC standard (2 GHz). Cette démonstration a permis de constater la robustesse de nos algorithmes, et de mesurer l'intérêt des utilisateurs pour ce type d'application.

À cette occasion, nous avons constaté que la segmentation de la main est une phase sensible : le contour obtenu est parfois trop imprécis, à cause notamment des variations de luminosité, ce qui affecte l'extraction de caractéristiques et la reconnaissance de postures (basée sur le contour). Par ailleurs, notre système souffre de certaines limitations, notamment concernant le vocabulaire utilisé pour la reconnaissance de postures, qui est limité à 11 gestes, et le suivi avec le modèle squelettique, qui ne permet pas de détecter les doigts collés ou repliés, ni de savoir quel doigt est plié et quel doigt est tendu.

PERSPECTIVES

Les limitations de notre système amènent à réfléchir sur des perspectives d'améliorations, que l'on peut décrire à chaque niveau du système :

SEGMENTATION : une amélioration possible serait de combiner l'information de couleur avec d'autres méthodes telles que la détection de contour ou de mouvement, ou l'information de profondeur qui peut être calculée grâce à la vision stéréoscopique. Toutefois, un compromis entre les performances et la complexité doit être trouvé.

MODÉLISATION DE LA MAIN : la modélisation 3D doit permettre de retrouver la configuration exacte de la main, notamment lorsque les doigts sont collés ou repliés. Les connaissances a priori sur la morphologie de la main pourrait être mieux exploitées. De plus, un recalage du modèle 3D en le projetant dans les images apporterait des informations supplémentaires.

GESTES DYNAMIQUES : la reconnaissance de gestes dynamiques représente l'évolution logique des travaux présentés ici, avec la reconnaissance des trajectoires des bouts des doigts et la reconnaissance de séquences de postures pour augmenter le nombre de gestes reconnus.

ASPECT ERGONOMIQUE : le développement d'applications basées sur ce type de système est nécessaire pour mieux prendre en compte les attentes des utilisateurs, et les limitations des méthodes actuelles. De plus, une réflexion est nécessaire sur la conception des interfaces graphiques, notamment d'un point de vue ergonomique, afin de tirer partie des possibilités d'interaction gestuelle.

GESTES BI-MANUELS : la prise en considération des deux mains permettrait d'augmenter fortement le nombre de gestes à reconnaître et donc les possibilités d'interaction. Toutefois, il serait nécessaire de bien distinguer les deux mains et de gérer les occultations entre elles.

ANNEXES

A

SOUSTRACTION DU FOND AVEC UN MÉLANGE DE GAUSSIENNES

Pour améliorer la segmentation par soustraction du fond, présentée au paragraphe 4.2.2.2, il est possible de réactualiser l'image de référence de façon non supervisée, en utilisant les propriétés statistiques des pixels. Le principe proposé par Stauffer et Grimson [122][123], et repris dans de nombreux travaux, consiste à modéliser l'histogramme temporel de chaque pixel par un mélange de gaussiennes. Dans ce mélange, les gaussiennes représentent soit le fond, soit les objets.

A.1 MODÉLISATION DES PIXELS PAR MÉLANGE DE GAUSSIENNES

L'algorithme proposé par Stauffer et Grimson [122][123] est une approximation « *on-line* » de l'algorithme EM [1], algorithme qui permet d'estimer les paramètres d'un mélange de gaussiennes à partir de données incomplètes [38]. L'algorithme de Stauffer traite directement la nouvelle valeur de chaque pixel, pour mettre à jour les paramètres du mélange de gaussiennes du pixel en question. Ainsi, il n'est plus nécessaire de stocker un historique de valeur pour chaque pixel.

Considérons l'historique temporel d'un pixel (x, y) de l'image I :

$$\{X_1, \ldots, X_t\} = \{I(x, y, i), 1 \leq i \leq t\}$$

Il est possible de construire son histogramme temporel (figure A.1). L'histogramme classique, pour une image, représente en abscisse les valeurs de 0 à 255, et en ordonnée le nombre de pixels prenant cette valeur. Pour l'histogramme temporel, le principe est le même, en remplaçant les valeurs des pixels d'une image par les valeurs d'un pixel au cours du temps.

Cet histogramme temporel est modélisé par un mélange de gaussiennes adaptatives, ce qui veut dire que les paramètres des gaussiennes sont actualisés au fil du temps, en prenant en compte les nouvelles valeurs du pixel. Plusieurs gaussiennes sont utilisées [2] afin de pouvoir modéliser les différents états du pixel.

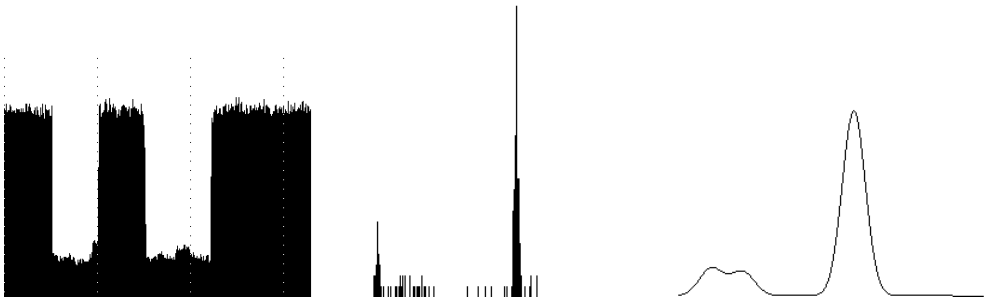
La probabilité d'observer la valeur du pixel courant X_t s'écrit comme la somme de K gaussiennes :

$$P(X_t) = \sum_{k=1}^{K} w_{k,t} * g(\mu_{k,t}, \sigma_{k,t}, X_t) \tag{A.1}$$

avec $g(\mu_{k,t}, \sigma_{k,t}, X_t)$ la densité de probabilité de la gaussienne k :

$$g(\mu_{k,t}, \sigma_{k,t}, X_t) = \frac{1}{\sqrt{2\pi}\sigma_{k,t}} \exp\left(-\frac{(X_t - \mu_{k,t})^2}{2\sigma_{k,t}^2}\right) \tag{A.2}$$

1. *Expectation-Maximisation*
2. en pratique un maximum de 5 gaussiennes est largement suffisant



(a) *historique temporel d'un pixel*

(b) *histogramme temporel du pixel*

(c) *mélange de gaussiennes associé*

FIGURE A.1 – *Modélisation de l'histogramme temporel d'un pixel par un mélange de gaussiennes.*

avec $\mu_{k,t}$ et $\sigma_{k,t}$ la moyenne et l'écart-type de la gaussienne, et $w_{k,t}$ le poids calculé pour chaque gaussienne à l'instant t.

Ces formules sont présentées pour le cas unidimensionnel (niveaux de gris) pour des raisons de simplicité, mais elles peuvent être facilement écrites pour le cas couleur (avec une gaussienne à trois dimensions, et une matrice de covariance).

A.2 PLUSIEURS GAUSSIENNES POUR LE FOND

Avec cette méthode, il n'est plus nécessaire de calculer explicitement l'image de référence. Il suffit de connaître la ou les gaussiennes qui correspondent au fond. Pour cela, on utilise le rapport poids/variance :

POIDS w : en supposant que le fond est plus souvent présent que les objets, la gaussienne du fond est celle de poids le plus fort.

VARIANCE σ : la valeur du fond étant pratiquement constante (sauf évolution de la luminosité), la variance de la gaussienne du fond est faible, alors que celle des objets est plus grande.

Après avoir trié les gaussiennes selon le rapport w/σ, on choisit la gaussienne de rapport le plus élevé comme étant la gaussienne de référence.

Il est possible d'associer plusieurs gaussiennes au fond, en prenant les B premières distributions telles que :

$$B = \arg\min_{b} \left(\sum_{k=1}^{b} w_k > T \right) \tag{A.3}$$

Cette formule permet de prendre en compte des distributions bimodales, lorsque le fond peut prendre plusieurs états : une fois les gaussiennes triées dans l'ordre du rapport w/σ, on choisit les B premières distributions telles que la somme des poids soit supérieure à T. Le paramètre T est donc une mesure de la proportion minimum de données que l'on veut associer au fond.

A.3 MISE À JOUR DES PARAMÈTRES

Pour chaque nouvelle valeur X_t du pixel, on cherche la gaussienne du mélange qui correspond au mieux à cette valeur, c'est-à-dire la gaussienne k telle que :

$$|\mu_k - X_t| \leq 2,5\,\sigma_k \tag{A.4}$$

Cette relation fournit une correspondance avec un seuillage par pixel et par distribution, c'est-à-dire un seuillage adapté au mélange de gaussiennes de chaque pixel, et où la correspondance avec une gaussienne dépend de la moyenne et de l'écart-type de cette gaussienne.

Afin d'éviter le risque de double correspondance, on prend en pratique la gaussienne qui minimise le rapport $|\mu_k - X_t|/\sigma_k$. Ainsi, on est sûr de prendre celle qui correspond le mieux.

Si une gaussienne γ correspond, sa moyenne et son écart-type sont mis à jour (les paramètres des autres gaussiennes restant inchangés), selon les relations suivantes[3] :

$$\mu_t = (1-\alpha)\mu_{t-1} + \alpha X_t \tag{A.5}$$

$$\sigma_t^2 = (1-\alpha)\sigma_{t-1}^2 + \alpha(X_t - \mu_t)^T(X_t - \mu_t) \tag{A.6}$$

Ces équations sont valables avec des paramètres scalaires, pour une vidéo en niveaux de gris, mais aussi dans le cas d'un espace couleur avec μ_t et X_t des vecteurs à trois composantes RGB, et σ_t^2 qui devient Σ_t, la matrice de covariance de dimension 3×3.

Si aucune correspondance n'est trouvée, ce qui est le cas au début de l'algorithme, on initialise une nouvelle gaussienne[4], avec pour moyenne X_t, un écart-type élevé et un poids faible. Ensuite, les poids sont mis à jour de la façon suivante :

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha.M_{k,t} \tag{A.7}$$

avec $M_{k,t} = 1$ pour la gaussienne qui correspond et 0 pour les autres. Enfin, les poids sont normalisés en divisant chaque $w_{k,t}$ par la somme des poids.

A.4 SUPPRESSION DES OMBRES

Il est possible d'adapter la méthode de suppression des ombres données par les équations 4.9, afin de vérifier si les pixels que nous avons segmentés sont des pixels du fond ombragés ou non. Cette méthode nécessite de réaliser le mélange de gaussiennes dans l'espace RGB.

Après calcul, nous obtenons les formules suivantes :

$$\alpha_i = \frac{I_R(i)\mu_R(i) + I_G(i)\mu_G(i) + I_B(i)\mu_B(i)}{(\mu_R(i))^2 + (\mu_G(i))^2 + (\mu_B(i))^2} \tag{A.8}$$

$$CD_i = \sqrt{\sum_{L\in[R,G,B]} \left(\frac{I_L(i) - \alpha_i\mu_L(i)}{\sigma_L(i)}\right)^2} \tag{A.9}$$

3. dans l'article de STAUFFER, ces mises à jour sont faites avec une variable ρ dérivée de α. Mais, de la même façon que pour KAEWTRAKULPONG ET BOWDEN [72], nos tests ont montré qu'il est plus pertinent de prendre $\rho = \alpha$.

4. ou, si elles sont déjà toutes initialisées, on remplace la gaussienne de poids le plus faible.

(a) *image 60* (b) *image 110*

(c) *image 150* (d) *image 210*

FIGURE A.2 – *Résultats de la segmentation d'objets par mélange de gaussiennes, sur la séquence « Hallmonitor ».*

avec $\mu_L(i)$ et $\sigma_L(i)$ la moyenne et l'écart-type de la composante $L \in [R, G, B]$ de la gaussienne du fond du pixel i.

En seuillant α_i et CD_i, on peut alors supprimer du masque les pixels correspondant à du fond ombragé ou illuminé. Le problème est de déterminer ces seuils, en fonction de la vidéo. Un autre problème est que, plus les valeurs des pixels sont sombres, plus elles sont proches de l'origine du repère RGB, et il devient difficile de distinguer les pixels du fond des autres. Il existe donc une valeur minimale de α_i en dessous de laquelle il vaut mieux ne rien faire.

La figure A.2 montre le résultat de la segmentation, avec le mélange de gaussiennes dans l'espace RGB, et la détection des ombres. Les résultats complets sont disponibles dans CONSEIL [23].

B

VISION STÉRÉOSCOPIQUE

L'objectif de ce chapitre est de présenter les bases de géométrie des caméras et de vision stéréoscopique, nécessaires à la bonne compréhension de ce manuscrit. Cette présentation est très succincte, pour plus de détails sur le sujet nous renvoyons à la littérature abondante sur ce domaine : les livres de référence de Faugeras [40] [41], Hartley et Zisserman [55], et Horaud et Monga [59] ; les thèses de Boufama [10] et Vezien [128] ; et les articles de revue faisant l'état de l'art du domaine, de Dhond et Aggarwal [39] pour les années 1980, et Brown *et al.* [16] pour les dernières avancées, concernant notamment les nouvelles techniques d'appariement, la problématique des occultations et les mises en oeuvre temps réel.

NOTATIONS

Dans ce chapitre, les notations suivantes sont utilisées :

$\mathcal{R}$	le repère de l'espace,
$\mathcal{R}_i$	le repère de l'image, centré sur le coin supérieur gauche,
$\mathcal{R}_c$	le repère de la caméra, centré sur F, le centre de la caméra,
f	la distance focale,
O	l'intersection de l'axe optique avec le plan de l'image, de coordonnées (u_0, v_0) dans le repère de l'image,
M	un point de l'espace, de coordonnées (X, Y, Z) dans $\mathcal{R}$,
p	le projeté de M dans l'image, de coordonnées (u, v) dans $\mathcal{R}_i$,
(R, t)	la matrice de rotation et le vecteur de translation entre $\mathcal{R}$ et $\mathcal{R}_c$

B.1 MODÈLE GÉOMÉTRIQUE DES CAMÉRAS

Une caméra réalise une projection de l'espace 3D vers un plan image 2D (figure B.1). Le modèle général est le modèle projectif. Il est représenté par la *matrice de projection perspective*, P, de dimensions 3×4. Cette matrice détermine la projection du point de l'espace M en un point de l'image p :

$$\mathrm{p} = \mathrm{PM} \quad \text{avec} \quad \mathrm{P} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \tag{B.1}$$

Le modèle projectif peut être décomposé en deux transformations :

A. La projection perspective : projection d'un point de l'espace 3D en un point de l'image 2D, dans le repère caméra.

B. La transformation du repère caméra au repère image (rotation et translation).

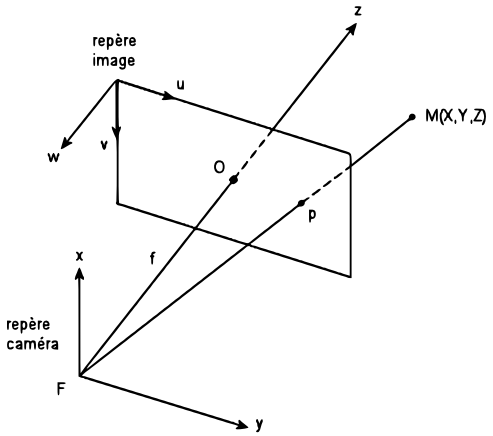


FIGURE B.1 – *Modèle de caméra, projection perspective.*

La matrice de projection peut donc se décomposer de la façon suivante :

$$\mathrm{P} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.2}$$

Ces matrices sont composées :
- des *paramètres intrinsèques* : α_u et α_v (facteurs d'échelle), u_0 et v_0 (coordonnées du centre de projection),
- des *paramètres extrinsèques* : r_{ij} et t_k (rotation et translation).

Ces paramètres peuvent être estimés par une procédure de calibration, présentée à la section B.2.

B.1.1 *Distorsion radiale*

L'utilisation d'optiques avec un angle de vue important introduit des distorsions dans l'image. En pratique, c'est la distorsion radiale qui a le plus d'importance. Elle résulte de défauts dans le système optique. Les causes de la distorsion radiale sont généralement modélisées par 5 paramètres :

$$\begin{cases} \delta_x(x,y) & = k_1 x(x^2+y^2) + p_1(3x^2+y^2) + 2p_2xy + s_1(x^2+y^2) \\ \delta_y(x,y) & = k_1 y(x^2+y^2) + p_2(x^2+3y^2) + 2p_1xy + s_2(x^2+y^2) \end{cases} \tag{B.3}$$

Ces paramètres sont estimés au cours de la calibration.

B.1.2 *Géométrie projective et coordonnées homogènes*

La géométrie euclidienne, qui permet de décrire angles et formes des objets, présente un aspect gênant : en 2D, deux droites s'interceptent toujours en un point, sauf les droites parallèles. On dit qu'elles se rencontrent à l'infini, mais l'infini n'existe pas.

L'espace projectif est un espace euclidien auquel on ajoute les points à l'infini. L'espace projectif est adapté à la vision par ordinateur, pour représenter l'espace 3D et la projection 2D des images.

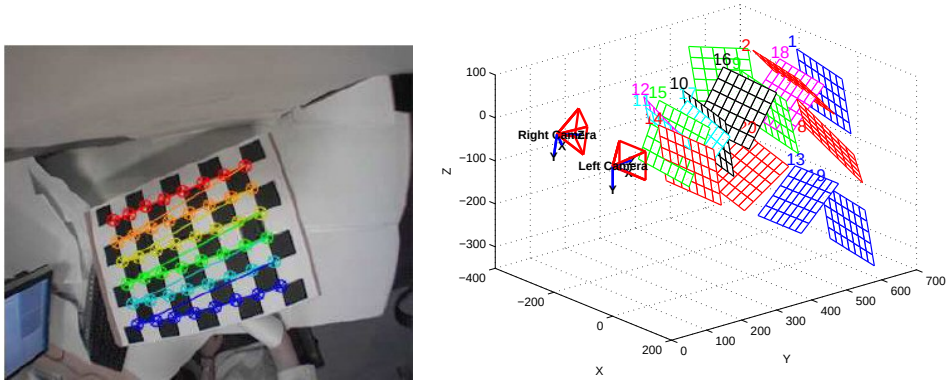
Dans un espace euclidien 2D, on représente un point par deux coordonnées (x, y). En ajoutant une coordonnée, on obtient le même point en *coordonnées homogènes* $(x, y, 1)$. Tous les points $(kx, ky, k), \forall k$ représentent le même point. $(x, y, 0)$ représente le *point à l'infini*. Ainsi l'espace euclidien $\mathbb{R}^n$ peut être étendu à un espace projectif $\mathbb{P}^n$ en représentant les points par un vecteur en coordonnées homogènes.

$$\begin{aligned} &\mathbb{R}^2 \rightarrow \mathbb{P}^2 \\ &(x, y) \rightarrow \begin{cases} (x, y, 1) & \equiv (kx, ky, k), \forall k \\ (x, y, 0) & : \text{points à l'infini} \end{cases} \end{aligned} \tag{B.4}$$

B.2 CALIBRATION

Dans le cas de la projection perspective, la calibration consiste à trouver à la fois les paramètres intrinsèques et extrinsèques de la caméra. La calibration s'effectue à partir de correspondances entre des points 3D et leurs projections, dont les coordonnées sont connues. Les équations forment alors un système linéaire dont la résolution fournit les paramètres. Il est nécessaire de connaître 6 points non coplanaires. En pratique, on prend plus de points et on effectue une minimisation pour avoir une meilleure précision. Pour connaître les coordonnées de ces points, on utilise une mire de calibration (figure B.2a).

La calibration est un problème fondamental en vision par ordinateur car la précision de la reconstruction 3D dépend de la bonne estimation des paramètres de calibration. De nombreuses recherches ont été effectuées sur l'autocalibration, qui consiste à calculer les paramètres à partir de points détectés dans la scène, sans utiliser de mire, ce qui permet d'automatiser la calibration.



(a) *Mire de calibration* (b) *Exemple de calibration stéréo*

FIGURE B.2 – *Calibration : (a) mire de calibration et détection automatique des points de la mire, et (b) Exemple de calibration stéréo avec la toolbox pour Matlab : les différentes positions de la mire sont représentées dans le repère 3D.*

Pour effectuer la calibration à partir d'images, une « Toolbox » pour Matlab (figure B.2b) est disponible en ligne [1], mais elle nécessite de déterminer manuellement les quatres coins de la mire, ce qui est vite fastidieux avec deux caméras et 10 à 15 images pour chaque caméra.

Il existe aussi une méthode automatique dans la bibliothèque OpenCV [2], mais elle est moins précise, principalement à cause de la détection automatique des coins de la mire qui donne parfois de mauvais résultats.

B.3 VISION STÉRÉOSCOPIQUE

Selon le principe de la projection perspective, un point d'une image correspond à une droite de l'espace, autrement dit tous les points de cette droite se projettent en un point unique de l'image. Par conséquent, il n'est pas possible de retrouver la structure 3D d'une scène à partir d'une seule image. C'est possible en utilisant des informations géométriques sur la scène, comme les points ou les lignes de fuite. Par contre, de façon analogue à la vision humaine, l'utilisation de deux ou plusieurs vues permet de calculer la structure 3D d'une scène. À partir des projections dans les deux images, il est possible, par triangulation, de retrouver les coordonnées du point en trois dimensions, c'est la stéréoscopie passive. Une autre solution est la stéréoscopie active, qui utilise des capteurs spécifiques comme un faisceau laser.

Un système stéréoscopique nécessite généralement une calibration, afin de connaître la matrice de transformation entre le repère caméra gauche et le repère caméra droite. Dans le cas où les paramètres internes des caméras sont connus mais pas la transformation entre les deux caméras, il faut estimer la *matrice essentielle*, et on obtient alors une *reconstruction euclidienne*. Si aucun calibrage n'a été fait, il faut estimer la *matrice fondamentale* pour obtenir une *reconstruction projective tridimensionnelle*. Connaissant cette matrice pour les deux caméras, la reconstruction consiste en une triangulation.

Les deux caméras [3] sont associées à une matrice de projection, respectivement P et P′. Un point M de l'espace projectif se projette respectivement en p et p′ (figure B.3) :

$$\begin{aligned} p &= PM \\ p' &= P'M \end{aligned} \tag{B.5}$$

Pour faire le processus inverse et calculer la position 3D d'un point de l'espace à partir de ses projections dans les deux caméras, il faut connaître la correspondance les deux projections, p et p′. On parle d'*appariement*, ou de *mise en correspondance*. Le point 3D se situe alors à l'intersection des deux droites de vues (Fp) et (F′p′), F et F′ étant les foyers des deux caméras.

En pratique ces deux droites ne s'intersectent pas forcément, du fait des erreurs de calculs. On procède donc à une minimisation de la distance entre les deux droites.

Les grandes étapes nécessaires pour reconstruire une scène en 3D sont les suivantes :

1. http:www.vision.caltech.edu/bouguetj/
2. *Open Computer Vision Library* : http://www.intel.com/research/mrl/research/opencv/
3. Par abus de langage, on parle fréquemment des caméras gauche et droite.

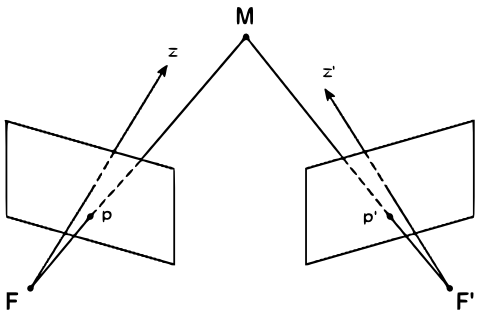


FIGURE B.3 – *Vision stéréoscopique*

EXTRACTION DE PRIMITIVES : points d'intérêts, segments de droites, courbes... les points étant les plus utilisés.

APPARIEMENT : mise en correspondance des primitives.

RECONSTRUCTION : calcul de la position 3D des primitives.

B.3.1 *Matrice fondamentale*

Elle représente les contraintes entre les points projetés dans deux images, p et p′, projections d'un même point de l'espace M. Une paire de points appariés satisfait la relation :

$$\mathrm{p'}^{T}\,\mathrm{F}\,\mathrm{p} = 0 \tag{B.6}$$

avec F la *matrice fondamentale*, de dimensions 3×3 et de rang 2. Une paire de caméras détermine une matrice fondamentale unique. Inversement, la matrice fondamentale peut être calculée à partir d'un ensemble de points appariés ce qui permet d'obtenir une reconstruction 3D projective.

B.3.2 *Relation gauche-droite*

Pour un système stéréoscopique, la calibration permet de connaître la relation géométrique entre le repère caméra gauche et le repère caméra droite. Cette relation est décrite par la matrice A_s, composée d'une matrice de rotation et d'une matrice de translation. Avec A_1 et A_2 les matrices des paramètres extrinsèques des deux caméras, on a :

$$A_s = A_1^{-1} A_2 \tag{B.7}$$

Ainsi, pour un point M de la scène, de coordonnées (X, Y, Z) dans le repère gauche, et (X', Y', Z') dans le repère droite, on a :

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathrm{A}_s \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} \tag{B.8}$$

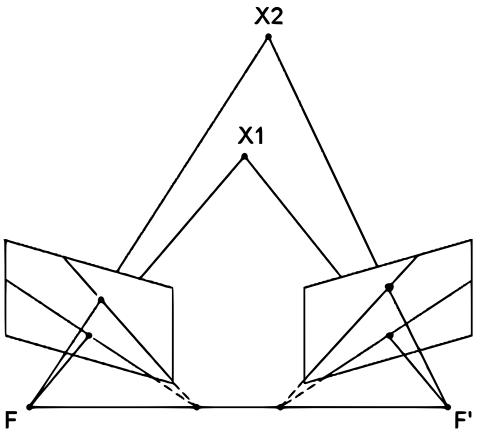


FIGURE B.4 – *Droites épipolaires*

B.3.3 *Géométrie épipolaire*

L'image de la droite de vue (Fp) dans l'autre caméra est la droite épipolaire, et réciproquement (figure B.4). Les droites épipolaires d'une caméra s'intersectent en un point appelé épipôle. Ce point est situé sur la droite (FF′) entre les centres des deux caméras.

Les droites épipolaires se calculent avec la matrice fondamentale. La droite épipolaire l' (resp. l) est associée au point p (resp. p'), et s'obtient par :

$$l' = Fp \tag{B.9}$$

$$l = F^T p' \tag{B.10}$$

La *contrainte épipolaire* exprime le fait que le correspondant d'un point d'une image se trouve sur une droite dans l'autre image : c'est l'image de la droite de vue du premier point. Cette contrainte permet de restreindre la recherche de correspondants, en sachant que le correspondant d'un point d'une image se trouve sur une droite dans l'autre image : l'image de la droite de vue.

La géométrie épipolaire implique aussi d'autres contraintes :

CONTRAINTE D'UNICITÉ : chaque point de l'image admet au plus un correspondant dans l'autre image.

CONTRAINTE DE CONTINUITÉ : on suppose la profondeur de la scène continue.

CONTRAINTE D'ORDRE : l'ordre des points est conservé dans les images.

Ces contraintes permettent de faciliter la mise en correspondance. Toutefois, elles ne sont pas toujours respectées, notamment lors d'occultations ou de fortes variations de profondeur dans la scène.

B.3.4 *Images parallèles ou rectifiées*

Afin de faciliter la mise en correspondance et d'obtenir rapidement une carte de disparité, les systèmes stéréoscopiques sont généralement constitués de deux caméras proches et quasiment parallèles. Si elles ne sont pas parallèles, il est possible de rectifier les images. Dans ce cas, les droites épipolaires sont parallèles

et elles correspondent aux lignes de l'image (figure B.5). Les épipôles sont alors situés à l'infini.

Ce cas particulier permet de simplifier l'appariement qui peut se faire en temps réel avec une mesure de corrélation sur les lignes de l'image. On obtient ainsi une carte de disparité (figure B.6).

La disparité est inversement proportionnelle à la profondeur. En notant L la distance entre les centres optiques des caméras[4], f la distance focale, et D la distance entre les abscisses des deux points projetés, on a :

$$Z = \frac{Lf}{D} \tag{B.11}$$

Dans le cas où les coordonnées 3D de tous les points sont calculées, on obtient une reconstruction dense de la scène. Le cas des caméras parallèles a l'avantage de faciliter la mise en correspondance mais la précision de la reconstruction est en général moins bonne (l'angle entre les droites de vue étant faible). Dans le cas des caméras non parallèles, l'appariement est plus difficile, mais la précision est meilleure.

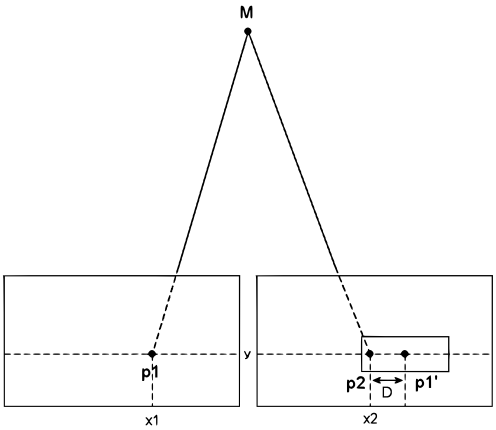


FIGURE B.5 – *Cas de deux images parallèles (rectifiées)*

(a) *Image gauche* (b) *Image droite* (c) *Carte de disparité*

FIGURE B.6 – *Exemple de carte de disparité avec la paire d'images Tsukuba.*

4. *baseline*

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

BIBLIOGRAPHIE

[1] K. Abe, H. Saito et S. Ozawa : Virtual 3D interface system via hand motion recognition from two cameras. *IEEE trans. on Systems, Man, and Cybernetics*, 32(4):536–540, juillet 2002.

[2] J. F. Abramatic, P. Letellier et M. Nadler : A narrow-band video communication system for the transmission of sign language over ordinary telephone lines. *In Image Sequences Processing and Dynamic Scene Analysis*, p. 314–336, 1983.

[3] T. Ahmad, C. J. Taylor, A. Lanitis et T. F. Cootes : Tracking and recognising hand gestures, using statistical shape models. *Image and Vision Computing*, 15(5):345–352, mai 1997.

[4] V. Athitsos et S. Sclaroff : An appearance-based framework for 3D hand shape classification and camera viewpoint estimation. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 40–45, mai 2002.

[5] V. Athitsos et S. Sclaroff : Estimating 3D hand pose from a cluttered image. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 2, juin 2003.

[6] P. Bertolino, G. Foret et D. Pellerin : Détection de personnes dans les vidéos pour leur immersion dans un espace virtuel. *In GRETSI, 18ème Colloque sur le Traitement du Signal et de l'Image*, 2001.

[7] J. Black et T. Ellis : Multi camera image tracking. *Image and Vision Computing*, 24(11):1256–1267, novembre 2006.

[8] M. J. Black et A. D. Jepson : Recognizing temporal trajectories using the condensation algorithm. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 16–21, 1998.

[9] A. F. Bobick et J. W. Davis : The recognition of human movement using temporal templates. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 23(3):257–267, mars 2001.

[10] B. Boufama : *Reconstruction Tridimensionnelle en Vision par Ordinateur : Cas des Caméras non Etalonnées*. Thèse de doctorat, INP de Grenoble, 1994.

[11] G. R. Bradski : Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2:15–26, 1998.

[12] A. Braffort : *Reconnaissance et compréhension de gestes, application à la langue des signes*. Thèse de doctorat, Université Paris-XI, 1996.

[13] F. Bérard : *Vision par ordinateur pour l'interaction homme-machine fortement couplée*. Thèse de doctorat, Université Joseph Fourier, 1999.

[14] F. Bérard, J. Coutaz et J. L. Crowley : Le tableau magique : un outil pour l'activité de réflexion. *In Proc. of the Ergonomie et Interactions Homme-Machine Conference (ERGO-IHM)*, 2000.

[15] L. Bretzner, I. Laptev et T. Lindeberg : Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2002.

[16] M. Z. Brown, D. Burschka et G. D. Hager : Advances in computationnal stereo. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, août 2003.

[17] C. Cadoz : Le geste canal de communication homme/machine - la communication instrumentale. *Techniques et Science Informatiques*, 13:31–61, 1994.

[18] A. Caplier, L. Bonnaud, S. Malassiotis et M. Strintzis : Comparison of 2D and 3D analysis for automated cued speech gesture recognition. *In SPECOM*, septembre 2004.

[19] D. Chai et K. N. Ngan : Face segmentation using skin-color map in videophone applications. *In IEEE Trans. on Circuits and Systems for Video Technology*, volume 9, p. 551–564, 1999.

[20] F.-S. Chen, C.-M. Fu et C.-L. Huang : Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vision Computing*, 21(8):745–758, août 2003.

[21] C. W. Chong, P. Raveendran et R. Mukundan : A comparative analysis of algorithms for fast computation of Zernike moments. *Pattern Recognition*, 36(3):731–742, mars 2003.

[22] D. Comaniciu, V. Ramesh et P. Meer : Real-time tracking of non-rigid objects using mean shift. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 2, p. 142–151, 2000.

[23] S. Conseil : Modélisation d'histogramme par mélange de gaussiennes pour la segmentation de séquences vidéo. Mémoire de D.E.A., INPG, 2003.

[24] S. Conseil, S. Bourennane et L. Martin : Suivi tridimensionnel en stéréovision. *In GRETSI, 20ème Colloque sur le Traitement du Signal et de l'Image*, 2005.

[25] S. Conseil, S. Bourennane et L. Martin : Three dimensional fingertip tracking in stereovision. *In Int. Conf. on Advanced Concepts for Intelligent Vision Systems*, volume 3708, p. 9–16. LNCS, septembre 2005.

[26] S. Conseil, S. Bourennane et L. Martin : Method of following hand movements in an image sequence. Brevet, janvier 2006. 06/100422 FR.

[27] S. Conseil, S. Bourennane et L. Martin : Comparison of Fourier descriptors and Hu moments for hand posture recognition. *In European Signal Processing Conference (EUSIPCO)*, 2007.

[28] T. F. Cootes, G. J. Edwards et C. J. Taylor : Active appearance models. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–684, juin 2001.

[29] T. F. Cootes, C. J. Taylor, D. H. Cooper et J. Graham : Active shape models-their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.

[30] T. R. Crimmins : A complete set of Fourier descriptors for two dimensional shape. *IEEE trans. on Systems, Man, and Cybernetics*, 6(121):848–855, 1982.

[31] J. Crowley, F. Berard et J. Coutaz : Finger tracking as an input device for augmented reality. *In IEEE Int. Work. on Automatic Face and Gesture Recognition*, p. 195–200, 1995.

[32] Y. Cui et J. Weng : Appearance-based hand sign recognition from intensity image sequences. *Computer Vision and Image Understanding*, 78:157–176, 2000.

[33] M. Daoudi, F. Ghorbel, A. Mokadem, O. Avaro et H. Sanson : Shape distances for contour tracking and motion estimation. *Pattern Recognition*, 32(7):1297–1306, juillet 1999.

[34] T. Darrell et A. Pentland : Space-time gestures. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, p. 335–340, 1993.

[35] J. Davis et M. Shah : Recognizing hand gestures. *In Proc. of the European Conference on Computer Vision*, p. 331–340, 1994.

[36] Q. Delamarre : *Suivi du mouvement d'objets articulés dans des séquences d'images vidéo*. Thèse de doctorat, Université de Nice - Sophia-Antipolis, 2003.

[37] Q. Delamarre et O. Faugeras : Finding pose of hand in video images : a stereo-based approach. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 585–590, 1998.

[38] A. P. Dempster, N. M. Laird et D. B. Rubin : Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[39] U. Dhond et J. Aggarwal : Structure from stereo - a review. *IEEE trans. on Systems, Man, and Cybernetics*, 19(6):1489–1510, 1989.

[40] O. Faugeras : *Three-dimensional computer vision : a geometric viewpoint*. MIT Press, 1993. ISBN 0-262-06158-9.

[41] O. Faugeras, Q.-T. Luong et T. Papadopoulou : *The geometry of multiple images : the laws that govern the formation of images of a scene and some of their applications*. MIT Press, 2001. ISBN 0-262-06220-8.

[42] W. T. Freeman, D. Anderson, P. Beardsley, C. Dodge, M. Roth, C. Weissman, W. Yerazunis, H. Kage, K. Kyuma, Y. Miyake et K. Tanaka : Computer vision for interactive computer graphics. *IEEE trans. on Computer Graphics and Applications*, 18(3):42–53, mai 1998.

[43] W. T. Freeman et M. Roth : Orientation histogram for hand gesture recognition. *In IEEE Int. Work. on Automatic Face and Gesture Recognition*, 1995.

[44] W. T. Freeman et C. D. Weissman : Television control by hand gestures. *In IEEE Int. Work. on Automatic Face and Gesture Recognition*, 1995.

[45] Y. Freund et R. E. Schapire : A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[46] D. M. Gavrila : The visual analysis of human movement : A survey. *Computer Vision and Image Understanding*, 73(1):82–98, janvier 1999.

[47] D. M. Gavrila et L. S. Davis : 3D model-based tracking of humans in action : a multiview approach. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 1996.

[48] F. Ghorbel : Towards a unitary formulation for invariant image description ; application to image coding. *Annals of Telecommunications*, 53 (3):143–153, mai 1998.

[49] F. Ghorbel et J. L. de Bougrenet de la Tocnaye : Automatic control of lamellibranch larva growth using contour invariant feature extraction. *Pattern Recognition*, 23:319–323, 1990.

[50] D. Hall, C. Gal, J. Martin, O. Chomat et J. L. Crowley : Magicboard : a contribution to an intelligent office environment. *Robotics and Autonomous Systems*, 35:211–220, 2001.

[51] J. Y. Han : Low-cost multi-touch sensing through frustrated total internal reflection. *In Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, 2005.

[52] P. R. G. Harding et T. Ellis : Recognizing hand gesture using Fourier descriptors. *In Proc. of the IEEE Int. Conf. on Pattern Recognition*, volume 3, p. 286–289, août 2004.

[53] I. Haritaoglu, D. Harwood et L. S. Davis : W4 : real-time surveillance of people and their activities. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 22(8):809–830, août 2000.

[54] P. A. Harling et A. D. N. Edwards : Hand tension as a gesture segmentation cue. *In Proc. of Gesture Workshop*, 1996.

[55] R. I. Hartley et A. Zisserman : *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. ISBN 0-521-54051-8.

[56] A. J. Heap et F. Samaria : Real-time hand tracking and gesture recognition using smart snakes. *In Proc. of Interface to Real and Virtual Worlds*, juin 1995.

[57] T. Heap et D. Hogg : Towards 3D hand tracking using a deformable model. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 140–145, octobre 1996.

[58] E-J. HOLDEN et R. OWENS : Recognising moving hand shapes. *In Proc. of the Int. Conf. on Image Analysis and Processing*, 2003.

[59] R. HORAUD et O. MONGA : *Vision par Ordinateur, Outils Fondamentaux*. Editions Hermès, 1995.

[60] T. HORPRASERT, D. HARWOOD et L. S. DAVIS : A statistical approach for real-time robust background subtraction and shadow detection. *In IEEE ICCV'99 Frame-Rate Workshop*, 1999.

[61] C-W. HSU, C-C. CHANG et C-J. LIN : A practical guide to support vector classification. Rapport technique, National Taiwan University, 2003.

[62] M-K. HU : Visual pattern recognition by moment invariants. *IEEE trans. on Information Theory*, 8:179–187, 1962.

[63] T. HUANG, V. PAVLOVIC et R. SHARMA : Gestural interface to a visual computing environment for molecular biologists. *In IEEE Int. Work. on Automatic Face and Gesture Recognition*, p. 30–35, 1996.

[64] Y. HUNG, Y. YANG, Y. CHEN, I. HSIEH et C. FUH : Free-hand pointer by use of an active stereo vision system. *In Proc. of the IEEE Int. Conf. on Pattern Recognition*, p. 1244–1246, 1998.

[65] S.-K. HWANG et W.-Y. KIM : A novel approach to the fast computation of Zernike moments. *Pattern Recognition*, 39(11):2065–2076, 2006.

[66] B. IONESCU, D. COQUIN, P. LAMBERT et V. BUZULOIU : Dynamic hand gesture recognition using the skeleton of the hand. *EURASIP Journal on Applied Signal Processing*, 13:2101–2109, 2005.

[67] M. ISARD et A. BLAKE : Condensation - conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29:5–28, 1998.

[68] Y. IWAI, K. WATANABE, Y. YAGI et M. YACHIDA : Gesture recognition using colored gloves. *In Proc. of the IEEE Int. Conf. on Pattern Recognition*, volume 1, p. 662–666, août 1996.

[69] N. JOJIC, T. HUANG, B. BRUMITT, B. MEYERS et S. HARRIS : Detection and estimation of pointing gestures in dense disparity maps. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2000.

[70] M. J. JONES et J.M. REHG : Statistical color models with application to skin detection. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, juin 1999.

[71] A. JUST, Y. RODRIGUEZ et S. MARCEL : Hand posture classification and recognition using the modified census transform. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 351–356, 2006.

[72] P. KAEWTRAKULPONG et R. BOWDEN : An improved adaptative background mixture model for real-time tracking with shadow detection. *In Proc., 2nd european workshop on advanced video based surveillance systems*, 2001.

[73] R. E. KALMAN : A new approach to linear filtering and prediction problems. *Journal of Basic Engineering, Transaction of the ASME*, 82:35–45, mars 1960.

[74] A. KENDON : *Current issues in the study of gesture*, p. 23–47. J. Nespoulous, P. Perron, and A. Lecours, Eds., Lawrence Erlbaum Associates, Hillsday, NJ, 1986.

[75] A. KHOTANZAD et Y. H. HONG : Invariant image recognition by Zernike moments. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 12(5): 489–497, mai 1990.

[76] M. KOHLER et S. SCHROTER : A survey of video-based gesture recognition - stereo and mono systems. Rapport technique, Fachbereich Informatik, University of Dortmund, 1998.

[77] M. KOLSCH et M. TURK : Analysis of rotational robustness of hand detection with Viola & Jones' method. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2004.

[78] M. KOLSCH et M. TURK : Robust hand detection. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2004.

[79] M. KOLSCH, M. TURK et T. HÖLLERER : Vision-based interfaces for mobility. *In Int. Conf. on Mobile and Ubiquitous Systems (MobiQuitous)*, 2004.

[80] W. W. KONG et S. RANGANATH : 3D hand trajectory recognition for signing exact English. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 535–540, mai 2004.

[81] J.J. KUCH et T.S. HUANG : Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration. *In Proc. of the IEEE Int. Conf. on Computer Vision*, p. 666–671, 1995.

[82] A. KUMAR et D. ZHANG : Personal recognition using hand shape and texture. *IEEE trans. on Image Processing*, 15(8):2454–2461, août 2006.

[83] S. KUMAR et C. SINGH : A study of Zernike moments and its use in Devanagari handwritten character recognition. *In Proc. of the Int. Conf. on Cognition and Recognition*, p. 514–520, 2005.

[84] C. LACHENAL : Interaction homme-machine et surfaces augmentées. Mémoire de D.E.A., INPG, 2000.

[85] J. LAVIOLA : A survey of hand posture and gesture recognition techniques and technology. Rapport technique CS-99-11, Department of Computer Science, Brown University, Providence, Rhode Island, 1999.

[86] J. LEE et T. KUNII : Model-based analysis of hand posture. *IEEE trans. on Computer Graphics and Applications*, 15:77–86, septembre 1995.

[87] A. LICSAR et T. SZIRANYI : User-adaptive hand gesture recognition system with interactive training. *Image and Vision Computing*, 23(12):1102–1114, novembre 2005.

[88] E. Lin, A. Cassidy, D. Hook, A. Baliga et T. Chen : Hand tracking using spatial gesture modeling and visual feedback for a virtual DJ system. *In Proc. of the IEEE Int. Conf. on Multimodal Interfaces*, page 197, sep 2002.

[89] R. Lockton et A. W. Fitzgibbon : Real-time gesture recognition using deterministic boosting. *In Proc. of the British Machine Vision Conference*, p. 817–826, 2002.

[90] J. Maclean, R. Herpers, C. Pantofaru, L. Wood, K. Derpanis, D. Topalovic et J. K. Tsotsos : Fast hand gesture recognition for real-time teleconferencing applications. *In IEEE ICCV Workshop on RATFG-RTS*, 2001.

[91] S. Marcel, O. Bernier, J.-E. Viallet et D. Collobert : Hand gesture recognition using input-output hidden markov models. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 456–461, 2000.

[92] J. Martin : *Reconnaissance de gestes en vision par ordinateur*. Thèse de doctorat, Institut National Polytechnique de Grenoble, 2000.

[93] J. Martin et J-B. Durand : Automatic handwriting gestures recognition using hidden Markov models. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2000.

[94] S. A. Mehdi et Y.N. Khan : Sign language recognition using sensor gloves. *In Proc. of the 9th Int. Conf. on Neural Information Processing*, volume 5, p. 2204–2206, novembre 2002.

[95] C. Mertz, J. L. Vinot et D. Etienne : Entre manipulation directe et reconnaissance de l'écriture : les gestes écologiques. *In Ergonomie et informatique Avancée, Interaction Homme-Machine Ergo-IHM*, p. 145–152, 2000.

[96] B. Moghaddam et A. Pentland : Probabilistic visual learning for object representation. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 19 (7):696–710, juillet 1997.

[97] A. Mokadem : *Distances entre formes, application au codage orienté objet*. Thèse de doctorat, Université de Compiègne, 1996.

[98] F. Mokhtarian et A. K. Mackworth : A theory of multiscale, curvature-based shape representation for planar curves. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 14(8):789–805, août 1992.

[99] K. Nickel, E. Scemann et R. Stiefelhagen : 3D-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 565–570, mai 2004.

[100] K. Oka, Y. Sato et H. Koike : Real-time fingertip tracking and gesture recognition. *IEEE trans. on Computer Graphics and Applications*, 22(6):64–71, novembre 2002.

[101] E. Ong et R. Bowden : Detection and segmentation of hand shapes using boosted classifiers. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2004.

[102] S. C. W. Ong et S. Ranganath : Automatic sign language analysis : A survey and the future beyond lexical meaning. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 27(6):873–891, juin 2005.

[103] N. Otsu : A threshold selection method from gray level histograms. *IEEE trans. on Systems, Man, and Cybernetics*, 9:62–66, mars 1979.

[104] H. Ouhaddi et P. Horain : Vers la modélisation du geste par la vision. *Traitement du Signal*, 16(1):15–29, 1999.

[105] V. Pavlovic, R. Sharma et T. Huang : Visual interpretation of hand gestures for Human-Computer Interaction : A review. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 19(7):677–692, juillet 1997.

[106] E. Persoon et K. S. Fu : Shape discrimination using Fourier descriptors. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 8(3):388–397, 1986.

[107] S. L. Phung, A. Bouzerdoum et D. Chai : Skin segmentation using color pixel classification : analysis and comparison. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 27(1):148–154, janvier 2005.

[108] R. Poppe et M. Poel : Comparison of silhouette shape descriptors for example-based human pose recovery. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 541–546, 2006.

[109] A. Prati, I. Mikic, M. M. Trivedi et R. Cucchiara : Detecting moving shadows : algorithms and evaluation. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 25(7):918–923, July 2003.

[110] F. K. H. Quek : Toward a vision-based hand gesture interface. *In Proc. of the Virtual reality software and technology conf.*, p. 17–31, 1994.

[111] J. Rehg et T. Kanade : Visual tracking of high DOF articulated structures : An application to human hand tracking. *In Proc. of the European Conference on Computer Vision*, volume II, p. 35–46, mai 1994.

[112] Y. Sato, K. Oka, H. Koike et Y. Nakanishi : Video-based tracking of user's motion for augmented desk interface. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2004.

[113] J. Schlenzig, E. Hunter et R. Jain : Vision-based hand gesture interpretation using recursive estimation. *In Proc. 28th Asilomar Conf. Signals, Systems, and Computer*, 1994.

[114] J. Segen et S. Kumar : Fast and accurate 3D gesture recognition interface. *In Proc. of the IEEE Int. Conf. on Pattern Recognition*, 1998.

[115] A. Shamaie et A. Sutherland : Hand tracking in bimanual movements. *Image and Vision Computing*, 23(13):1131–1149, 2005.

[116] J. Shawe-Taylor et N. Cristianini : *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[117] N. Shimada, K. Kimura et Y. Shirai : Real-time 3D hand posture estimation based on 2D appearance retrieval using monocular camera. *In IEEE ICCV Workshop on RATFG-RTS*, p. 23–30, 2001.

[118] N. SHIMADA, Y. SHIRAI, Y. KUNO et J. MIURA : Hand gesture estimation and model refinement using monocular camera - ambiguity limitations by inequality constraints. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 268–273, avril 1998.

[119] L. SIGAL, S. SCLAROFF et V. ATHITSOS : Skin color-based video segmentation under time-varying illumination. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 26(7):862–877, juillet 2004.

[120] T. STARNER et A. PENTLAND : Visual recognition of American sign language using hidden Markov models. *In IEEE Int. Work. on Automatic Face and Gesture Recognition*, p. 189–194, 1995.

[121] T. STARNER, J. WEAVER et A. PENTLAND : Real-time american sign language recognition using desk and wearable computer based video. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[122] C. STAUFFER et W. E. L GRIMSON : Adaptative background mixture models for real-time tracking. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 1999.

[123] C. STAUFFER et W. E. L GRIMSON : Learning patterns of activity using real-time tracking. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 22:747–757, août 2000.

[124] B. STENGER, P. MENDONÇA et R. CIPOLLA : Model-based 3D tracking of an articulated hand. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2001.

[125] D. J. STURMAN et D. ZELTZER : A survey of glove-based input. *IEEE trans. on Computer Graphics and Applications*, 14(1):30–39, janvier 1994.

[126] J. TRIESCH et C. von der MALSBURG : Robust classification of hand postures against complex backgrounds. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, p. 170–175, 1996.

[127] M. TURK et A. PENTLAND : Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:72–86, 1991.

[128] J. M. VEZIEN : *Techniques de reconstruction globale par analyse de paires d'images stéréoscopiques*. Thèse de doctorat, Université Paris 7, 1995.

[129] C. VOGLER et D. METAXAS : ASL recognition based on a coupling between HMMs and 3D motion analysis. *In Proc. of the IEEE Int. Conf. on Computer Vision*, p. 363–369, 1998.

[130] C. VOGLER et D. METAXAS : Parallel hidden markov models for american sign language recognition. *In Proc. of the IEEE Int. Conf. on Computer Vision*, p. 116–122, 1999.

[131] A. WABEL et M. VO : A multi-modal human-computer interface : Combination of gesture and speech recognition. *In Proc. of the Int. Conf. on Human Factors in Computing Systems (CHI)*, 1993.

[132] C. Wah Ng et S. Ranganath : Real-time gesture recognition system and application. *Image and Vision Computing*, 20(13-14):993–1007, décembre 2002.

[133] G. Welch et G. Bishop : An introduction to the Kalman filter. Rapport technique TR 95-041, University of North Carolina at Chapel Hill, 1995.

[134] P. Wellner : Interacting with paper on the DigitalDesk. *Communications of the ACM*, 36(7):87–96, 1993. ISSN 0001-0782.

[135] A. D. Wilson et A. F. Bobick : Parametric hidden Markov models for gesture recognition. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 21(9):884–900, septembre 1999.

[136] A. Wu, M. Shah et N. Da Vitoria Lobo : A virtual 3D blackboard : 3D finger tracking using a single camera. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2000.

[137] Y. Wu et T. S. Huang : Hand modeling, analysis, and recognition for vision-based human computer interaction. *IEEE Signal Processing Magazine*, 18 (3):51–60, mai 2001.

[138] Y. Wu, J. Lin et T. S. Huang : Analyzing and capturing articulated hand motion in image sequences. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 27(12):1910–1922, décembre 2005.

[139] H. Wuest : *Dynamic Tracking and Data Association in Image Sequences*. Thèse de doctorat, University of Mannheim (Germany), 2004.

[140] X. Yin et M. Xie : Hand gesture segmentation, recognition and application. *In Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, p. 438–443, 2001.

[141] E. Yoruk, E. Konukoglu, B. Sankur et J. Darbon : Shape-based hand recognition. *IEEE trans. on Image Processing*, 15(7):1803–1815, July 2006.

[142] D. Zhang et G. Lu : A comparative study on shape retrieval using Fourier descriptors with different shape signatures. *In Int. Conf. on Intelligent Multimedia and Distance Education*, 2001.

[143] X. Zhu, J. Yang et A. Waibel : Segmenting hands of arbitrary color. *In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2000.

[144] Y. Zhu, G. Xu et D. J. Kriegman : A real-time approach to the spotting, representation, and recognition of hand gestures for human-computer interaction. *Computer Vision and Image Understanding*, 85(3):189–208, mars 2002.

COLOPHON

Ce manuscrit de thèse a été rédigé avec LaTeX en utilisant la fonte *Palatino*. Le style typographique de ce manuscrit est basé sur le paquet « `classicthesis` [5] » réalisé par André Miede, disponible sur CTAN, et adapté à la typographie française grâce aux *Petites leçons de typographie* de Jacques André [6].

Version Finale du 8 avril 2008.

5. http://www.ctan.org/tex-archive/macros/latex/contrib/classicthesis/
6. http://jacques-andre.fr/faqtypo/

RÉSUMÉ

L'interprétation automatique de gestes basée sur la vision par ordinateur offre de nouvelles possibilités d'interaction avec l'ordinateur, plus naturelles et intuitives qu'avec les périphériques classiques. Cependant, le canal gestuel est un moyen de communication particulièrement riche et la main un objet articulé complexe. Ainsi, l'interaction homme-machine gestuelle constitue un axe de recherche particulièrement actif avec un potentiel applicatif important.

Dans ce contexte, notre travail a consisté à remplacer le fonctionnement d'un écran tactile par un système de vision stéréoscopique avec deux caméras. Ainsi, le problème fondamental a consisté à suivre, en temps réel, le mouvement de la main et des doigts à partir de leurs projections dans les images, avant d'en reconnaître la posture. Les contraintes industrielles qui ont guidé nos travaux nous ont orienté vers une approche par apparence, avec des hypothèses réduites afin que le système soit peu contraignant pour l'utilisateur. Les différentes étapes abordées concernent la détection de la main basée sur la couleur de peau, l'extraction de caractéristiques invariantes, la comparaison de descripteurs de forme pour la reconnaissance de postures 2D, et le suivi 3D du mouvement des doigts et de la main avec un modèle squelettique. Les algorithmes ont été évalués à l'aide d'une base originale de vidéos stéréoscopiques, montrant l'amélioration notable des solutions proposées. La robustesse du système a également été confrontée aux conditions réelles d'une démonstration publique.

MOTS CLÉS : interaction homme-machine, vision stéréoscopique, reconnaissance de gestes, modèle squelettique de la main, suivi 3D de la main.

THREE-DIMENSIONAL HAND TRACKING AND GESTURE RECOGNITION FOR HUMAN-COMPUTER INTERACTION

Automatic interpretation of gestures based on computer vision provides new possibilities of interaction with computers, more natural and intuitive than with classic devices. However, the gestural channel is a particularly rich means of communication and the hand is a complex articulated object. Thus, gesture-based human-computer interaction represents a particularly active research axis with an important potential of application.

In this context, our work consisted in replacing the mechanism of a tactile screen by a system based on stereoscopic vision with two video cameras. Thus, the central issue consisted in tracking in real-time the movement of the hand and the fingers based on their projections on images, and in recognizing the posture. The industrial constraints which guided our work lead us to an appearance-based approach, with reduced hypothesis so that the system is little restrictive for the user. The different steps we address concern hand detection based on skin color, extraction of invariant features, comparison of shape descriptors for 2D posture recognition, and 3D finger and hand tracking with a skeleton model. The performances of the algorithms was evaluated with an original database of stereoscopic video sequences, showing the noticeable improvement of the proposed solutions. The system's robustness was confronted to real conditions with a public demonstration.

KEYWORDS : human-computer interaction, stereoscopic vision, gesture recognition, skeleton hand model, 3D hand tracking.

INSTITUT FRESNEL, UMR 6133
D.U. de Saint Jérôme, 13397 Marseille Cedex 20 - FRANCE